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"**

A Procedure for Finding Initial Feasible Solutions on Cyclic Natural Gas Networks

Conrado Borraz-Sánchez
Roger Z. Ríos-Mercado

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

ABSTRACT

The fuel cost minimization problem on cyclic natural gas networks system is addressed. We provide and describe an efficient algorithm for achieving a feasible point inside a non-convex nonlinear compressor station domain. We work with several different types of topologies, many of those being cyclic structures. By tightening variable bounds at preprocessing, we propose a simple procedure for getting feasible solutions quickly. This procedure avoids the many numerical difficulties inherent to this very complex while treated with classical nonlinear programming techniques. A computational study revealed the effectiveness of the proposed procedure as it was able to deliver feasible solutions quickly to many instances with cyclic structures, and thus, outperforming previous approaches.

1. INTRODUCTION

Natural gas flow, driven by pressure, is transported through pipeline networks systems. During this phase, energy and pressure are lost due to both friction between the gas and the pipes' inner wall, and heat transfer between the gas an its environment. To keep the gas flowing through the system, it is necessary to periodically restore the gas (increase its presure), so compressor stations are installed in the network. These stations typically consume about 3 to 5% of the transported gas. This transportation cost is significant because the amount of gas being transported worldwide is huge. A typical network today might consist of thousands of pipes, dozens of stations, and many other devices, such as valves and regulators. Such a network may transport thousands of MMCFD (1 MMCFD = $10x10^6$ cubic feet per day) of gas. It is estimated [2] that the global optimization of operations can save at least 20% of the fuel consumed by the stations. Hence, the problem of finding out how to optimally operate the compressors driving the gas in a pipeline network becomes significantly important.

There are several variations of this problem depending on the modeling assumptions. Here, we address a problem where we consider two types of decision continuos variables: mass flow rate through each arc and pressure value at each node (Wu et al. [6]). So, the model is a nonlinear programming problem (NLP).

This problem has been addressed both as a (non-convex) NLP [3,4] and as a Mixed Integer Nonlinear Programming Problem (MINLP) [1]. In [3,4], for instance, a computational evaluation with a GRG code was reported. It was found how the algorithm had a relatively success on delivering local optimal solutions on instances with no cyclic network topologies. Similar results were observed in [1]. However, the main drawback of those approaches is on dealing with cyclic structures.

So the purpose of this work is to propose a simple and fast way of delivering good initial feasible solutions, aiming of course at cyclic network topologies. First we present the model and assumptions. Then, we derive both lower and upper bounds on mass flow rate, and suction and discharge pressures bounds of the compressor station. Then, describe an efficient procedure for finding a good initial feasible point on cyclic structures. The procedure consists of two phases. First a set of feasible flows is found and then an attempt is made to find a feasible set of pressure values (for the pre-specified flow). Last, a computational study over a large set of instances to assess the procedure performance is presented. The results were outstanding as it was found that the procedure outperformed former approaches and delivered feasible solutions very quickly over all cyclic instances tested.

2. MODEL DESCRIPTION

Assumptions: The following assumptions are made for solving this problem:

- We assume that the problem is in steady state. This is, our model will provide solutions 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 the 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 station are feed with some of the fuel driven through the pipelines, and for sustaining this assumption we consider the cost of this consumption as an extra (opportunity) cost in our model, that represent the amount we would spend if we had to buy the fuel from third parties.

- The network is directed. Each arc in the network has a pre-specified direction.
- The problem is deterministic. Each parameter is assumed known with certainty.

The NPL 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 pipelines arcs
Ac: Set of compressor stations arcs
A: Set of all arcs in the network; $A = Ap \cup Ac$
U_{ij}: Arc capacity of pipeline (i,j); (i,j) ε Ap
R_{ij}: Resistance of pipeline (i,j); (i,j) ε Ap
P_i^L, P_i^U: Pressure lower and upper limits at each node; i ε V
B_i: Net mass flow rate at node i; iεN. $B_i>0$ if iεVs, $B_i<0$ if iεVd, $B_i=0$ otherwise

Variables:

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

Formulation:

Minimize $\sum_{(i,j)\varepsilon Ac} g_{(i,j)}(x_{ij}, p_i, p_j)$ (1a)

$\sum_{\{j \mid (i,j)\,\varepsilon A\}} x_{ij} - \sum_{\{j \mid (i,j)\varepsilon A\}} x_{ji} = B_i \quad i\ \varepsilon\ V$ (1b)

$x_{ij} \leq U_{ij}$ (i,j) ε Ap (1c)

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

$p_i^L \leq p_i \leq p_i^U$ i ε V (1e)

$(x_{ij}, p_i, p_j)\ \varepsilon\ D_{ij}$ (i,j) ε Ac (1f)

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

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

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

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

$$S_{ij}^{\ L} \le s_{ij} \le S_{ij}^{\ U} \tag{3}$$

$$R^L \le \frac{q_{ij}}{s_{ij}} \le R^U \tag{4}$$

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:

$$h_{ij} = \frac{ZRT_s}{m}\left[\left(\frac{p_i}{p_j}\right)^m - 1\right] \tag{5}$$

and

$$q_{ij} = ZRT_s\frac{x_{ij}}{p_i} \tag{6}$$

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 square 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_{ji}	Inlet volumetric flow rate in compressor (i,j); (i,j) ∈ Ac
h_{ji}	Adiabatic head compressor (i,j); (i,j) ∈ Ac
s_{ji}	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.

For a detailed explanation about centrifugal compressor station and previous work, see Ríos-Mercado [5] or Wu et al. [6].

3. PRE-PROCESSING

When we first attempted to transport mass flow rate on a compressor arc, we noticed that exists both lower and upper bounds on mass flow rate, and suction and discharge pressures in the compressor station domain. This motivated our work in a preprocessing technique.

In short, this pre-processing techniques can be defined as elementary operations that tightens both lower and upper bounds on mass flow rate, and suction and discharge pressures in constraint (1f), where D_{ij} represents the feasible operating domain for compressor station (i,j). Hence, we define both mass flow, and inlet and outlet pressures on each compressor arc into feasible bounds that may lead to better algorithmic properties before attempting to solve it. By applying the pre-processing technique one expects that feasible solutions can be found more efficiently.

- **Lower and upper bounds on mass flow rate:** Bounds calculated are based joining equations (1e), (1g), (1f) and (6) for reducing the feasible region or search space, preventing the algorithm to examine boundless domains on each compressor arc (depends on compressor type).

$$x_{ij}^L = \frac{p_i^L q_{ij}^L}{ZRT_s} \;, \quad x_{ij}^U = \frac{p_i^U q_{ij}^U}{ZRT_s} \qquad (7)$$

Let x_{ij}^L and x_{ij}^U be the lower and upper bound on mass flow rate of the compressor arc, respectively. By (7) then

$$x_{ij}^L \leq x_{ij} \leq x_{ij}^U \qquad (i,j) \; \varepsilon \; Ac$$

- **Limits of pressure in each node joining compressor arc:** To deal with this we sort to assigning initial values selecting values within limits of pressure in each node. We calculate the lower and upper bound pressures joining equations (2), (3), (4) and (5). It is possible to lighten the pressure limits as follows.

Let $$\Phi\left(\frac{q_{ij}}{s_{ij}}\right) = A_H + B_H\left(\frac{q_{ij}}{s_{ij}}\right) + C_H\left(\frac{q_{ij}}{s_{ij}}\right)^2 + D_H\left(\frac{q_{ij}}{s_{ij}}\right)^3 \tag{8}$$

by (3), (4) and (8), it follows that the adiabatic head h_{ij} must satisfy

$$h_{ij}^L = \left(s_{ij}^L\right)^2 \Phi\left(R_{ij}^U\right) \tag{9}$$

and

$$h_{ij}^U = \left(s_{ij}^U\right)^2 \Phi\left(R_{ij}^L\right) \tag{10}$$

Let h_{ij}^L and h_{ij}^U be the lower and upper bound the adiabatic head, respectively.

Thus, by (1e), (5), (9), and (10) we get

$$p_{i2}^L = p_j^L \left[\frac{\left(mh_{ij}^U\right)}{ZRT_s} + 1\right]^{-\left(\frac{1}{m}\right)} \tag{11}$$

and

$$p_{i2}^U = p_j^U \left[\frac{\left(mh_{ij}^L\right)}{ZRT_s} + 1\right]^{-\left(\frac{1}{m}\right)} \tag{12}$$

so $\max\{ p_i^L, p_{i2}^L \} \leq p_i \leq \min\{ p_i^U, p_{i2}^U \}$.

4. PROCEDURE DESCRIPTION

In this part, we design a search algorithm for finding an initial feasible solution for the fuel cost minimization problem on cyclic networks systems (Figure 1). This algorithm consists of two phases or procedures. The first procedure makes use of the preprocessing technique to find a feasible flow at each arc on the net. Our procedure for constructing the feasible flow, utilizes a path from node j to node i (where $j \varepsilon V | b(j)<0$ and $i \varepsilon V | b(j)>0\}$. The second procedure also makes use of the preprocessing technique to find feasible pressures at each node on the net.

```
SEARCH ALGORITHM
begin
      Global INPUT: G=(V,A), where A={Ad, Ac}. B(i) sources value ∀iεV
      {Properties of natural gas pipeline network system}
      PROCEDURE-1 {Find feasible flows}
      PROCEDURE-2 {Find feasible pressures}
end
STOP
```

Figure 1. Search Algorithm for finding an initial feasible solution on cyclic structures.

Basically, in PROCEDURE-1 the algorithm finds a feasible set of flows or indicates infeasible solution basing on all set of paths of each node sink to each node supply. To do this, the procedure consists of three phases:

- Step 1 (Pre-processing): In this step, lower and upper flow bounds on each compressor arc are tighten.
- Step 2 (Path selection): Here, for a given pair (s,t) of supply and demand nodes, respectively, it finds all possible paths between them. Note that this number is relatively small since we are dealing with networks with only a small number of cycles. If the path is not unique, select a path containing a compressor arc, else choose one at random.
- Step 3 (Flow assignment): This phase assigns the flow mass rate on the path just found by taking into account the mass balance at the end nodes and the residual capacity of each arc along the path. Update flows and residual network.

The algorithm repeats the second and third step until all flow has been assigned. Pseudo-code of this is shown in Figure 3 in the Appendix.

Then, PROCEDURE-2 begins by assigning an initial pressure to a reference node $r \mid r \varepsilon V$. Next, by looking at adjacent arcs, it assigns first pressure values at nodes belonging to pipeline arcs (because this is uniquely determined by an equality constraint). Then it assigns a suction or discharge pressure at each adjacent compressor arc. The step is repeated iteratively until all set of pressures are found or an infeasibility stopping condition is met. In the latter case, the procedure restarts. Pseudocode of this is shown in Figure 4 in the Appendix.

5. COMPUTATIONAL EVALUATION

In order to assess the effectiveness of proposed procedure, we apply the search algorithm under different scenarios with different kinds of topologies. There are many types of topologies: (a) simple or gun-barrel, (b) tree, and (c) cyclic. Our evaluation is based on a database developed by Villalobos-Morales et al. [7]. For example, in Figure 2, a striped node represents a supply point, a black node (shown with an outgoing arrow next to it) represents a demand point, and a white node is a transshipment node. A single directed arc joining two nodes represents a pipeline, and a directed arc with a black trapezoid joining two nodes represents a compressor arc.

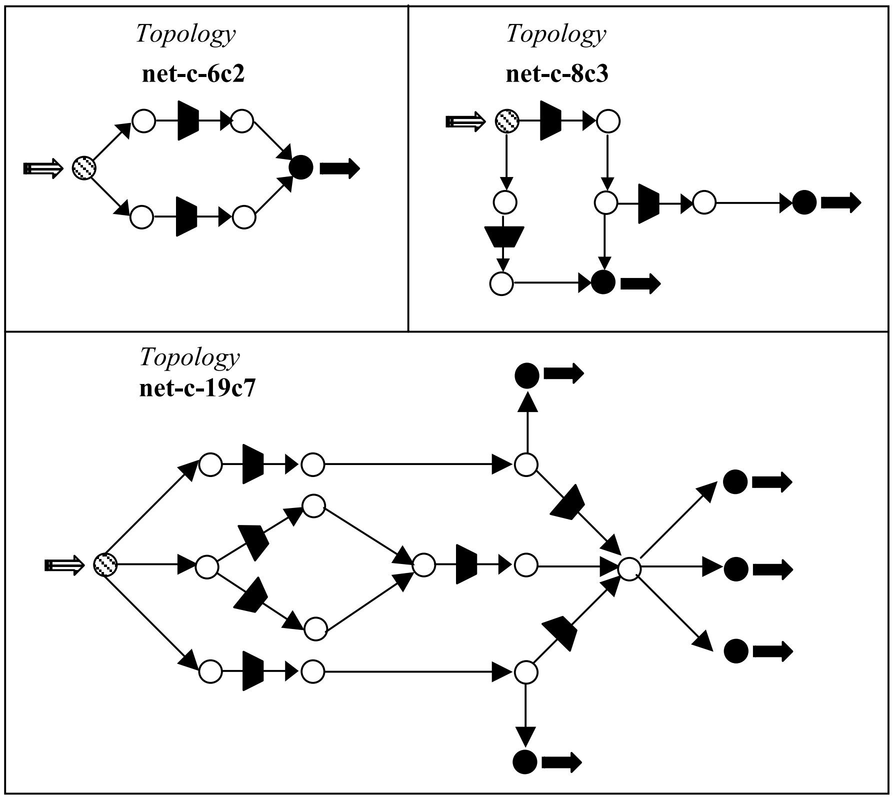

Figure 2. Examples of topologies type c used into database.

Recall that the motivation of this work stems from the fact that previous works have failed on obtaining feasible solutions on topologies with cyclic structures (type c). Our procedure was coded in C++, and run on a Sun Ultra 10 under Solaris 7.

The results are shown in Table 1. The instance tested are shown in the first column. The features of the topology are shown in the second column. The third column shows the CPU times used for finding a feasible solution. An (*) denotes new topologies with special structures added into this database.

The first thing to notice is that a feasible solution was successfully achieved in each instance tested. This shows the effectiveness of this proposed algorithm for finding feasible solutions. We also notice the very low CPU times.

Instance tested	Features: Node, Compressor, Compressor Type	CPU Time (sec)
net-a-8c3	Small Gun-barrel: 8 , 3, Type-4	0.03
net-b-10c3-C1	Small Tree: 10 , 3, Type-1	0.07
net-b-11c4	Medium Tree: 11 , 4, Type-4	0.07
net-b-15c6	Medium Tree: 15 , 6, Type-4	0.06
net-b-41c12	Large Tree: 41 , 12, Type-4	0.09
net-b-43c13	Large Tree: 43 ,13, Type-4	0.09
net-b-39c14	Large Tree: 39 ,14, Type-4	0.05
net-b-50c15	Large Tree: 50 ,15, Type-4	0.06
net-c-6c2-C1	Small Cyclic Structure: 6, 2, Type-1	0.06
net-c-6c2-C2	Small Cyclic Structure: 6, 2, Type-2	0.05
net-c-6c2-C3	Small Cyclic Structure: 6, 2, Type-3	0.06
net-c-6c2-C4	Small Cyclic Structure: 6, 2, Type-4	0.06
net-c-8c3-C1	Small Cyclic Structure: 8, 3, Type-1	0.06
net-c-8c3-C3	Small Cyclic Structure: 8, 3, Type-3	0.04
net-c-8c3-C4	Small Cyclic Structure: 8, 3, Type-4	0.05
net-c-8c3-C5	Small Cyclic Structure: 8, 3, Type-5	0.05
net-c-8c3-C7	Small Cyclic Structure: 8, 3, Type-7	0.04
net-c-10c3	Small Cyclic Structure: 10, 3, Type-4	0.04
net-c-13c5	Medium Cyclic Structure: 13, 5, Type-4	0.04
net-c-15c5	Medium Cyclic Structure: 15, 5, Type-4	0.05
net-c-17c6	Medium Cyclic Structure: 17, 6, Type-4	0.02
net-c-19c7	Large Cyclic Structure: 19, 7, Type-4	0.04
net-c-45c16	Huge Cyclic Structure: 45, 16, Type-4	0.17
net-c-50c19	Huge Cyclic Structure: 50, 19, Type-4	0.23
net-b-10c2(*)	Small Tree: 10, 2, Type-4	0.07
net-b-12c4(*)	Medium Tree: 12, 4, Type-4	0.02
net-b-13c5(*)	Medium Tree: 13, 5, Type-4	0.04
net-c-16c10(*)	Large Cyclic Structure: 16, 10, Type-4	0.07

Table 1. Results of instances tested and computational behavior.

6. CONCLUSIONS

Our preliminary computational experiments showed that it is evident the tremendous positive impact that the preprocessing technique had in the problem addressed. The application of this technique not only found feasible solutions in the entire database, but also reduced the resources (computational time) used by the computer. This represents a significant contribution, particularly when dealing with cyclic structures where previous approaches had failed.

This is an ongoing research. We are still working on local search heuristics to improve the initial feasible solution obtained in quest of a local optimal solution. All programs and files used in this work are available from the authors.

Acknowledgments: This research is supported by the Mexican National Council for Science and Technology (CONACYT grant J33187-A), and Universidad Autónoma de Nuevo León under its Scientific and Technological Research Support Program (PAICYT grants CA555-01 and CA763-02).

REFERENCES

[1] D. Cobos-Zaleta and R. Z. Ríos-Mercado. A MINLP model for minimizing fuel consumption on natural gas pipeline networks. In *Proceedings of the XI-Latin-Ibero-American Conference on Operations Research,* paper A48-01, pages 1-9. Concepción, Chile, October 2002.

[2] D. W. Schroeder. Hydraulic analysis in the natural gas industry. In J. J.-W. Chen and A. Mital (editors), *Advances in Industrial Engineering Applications and Practice I*, pages 960-965, International Journal of Industrial Engineering, Cincinnati, USA, 1996.

[3] H. J. Flores-Villarreal and R. Z. Ríos-Mercado. Computational experience with a GRG method for minimizing fuel consumption on cyclic natural gas networks. In N. Mastorakis, G. Antoniou, C. Manikopoulos, V. Mladenov, Z. Boikovic, and A. Varonides (editors), *Proceedings of the 7th WSEAS International Conference on Systems,* paper 457-146. Corfu, Greece, July 2003.

[4] H. J. Flores-Villarreal and R. Z. Ríos-Mercado. Efficient operation of natural gas pipeline networks: Computational finding of high quality solutions. In *Proceedings of the International Applied Business Research Conference,* paper 352, pages1-7. Acapulco, México, March 2003.

[5] R. Z. Ríos-Mercado. 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, USA, 2002.

[6] S. Wu, R. Z. Ríos-Mercado, E. A. Boyd, and L. R. Scott. Model relaxations for the fuel cost minimization of steady-state gas pipeline networks. *Mathematical and Computer Modeling,* 31(2-3): 197-220, 2000.

[7] Y. Villalobos-Morales, D. Cobos-Zaleta, H. J. Flores-Villarreal, C. Borraz-Sánchez, and R. Z. Ríos-Mercado. On NLP and MINLP formulations and preprocessing for fuel cost minimization of natural gas transmission networks. In *Proceedings of the 2003 NSF Design, Service and Manufacturing Grantees and Research Conference.* Birmingham, Alabama, USA, January 2003.

APPENDIX

Here we show pseudo-code for procedures 1 and 2 (depicted in Figure 1) for finding both feasible flows and pressures.

```
PROCEDURE-1 {Quest of feasible flows}
begin
0:   Input: Set PATHS from each node jεV│B(j)<0 to each node iεV│B(i)>0.
     Hint: Feasible_Paths_ALGORITHM (developed by C. Borraz-Sánchez, 2003).
     Let Γ(i ) = number of paths found for each sink node { iεV│B(i )<0 }
1:   PREPROCESSING–PHASE
     begin
           Building lower and upper bounds of flow mass at each compressor arc, and assign lower bound into whole paths
           that containing this compressor arc.
           Building limits of suction and discharge pressures at each node joining a compressor arc for using in
           PROCEDURE-2.
            For ∀(i,j)εA Do
               If node i has one outgoing arc Then
                        Assign B(i) source on (i,j)εA
               end
           end
     end
2:  While ∃ some path where node iεV│B(i)<0 Do
    begin
3:        THE BEST PATH SELECTING–PHASE
          begin
               If Γ(i )=2 Then take path with lower source value
               If Γ(i )>2 Then take any path
               If ∃ some Γ(λ)=1 where λεV Then take the only one path for this sink node
          end
4:        Eliminate any path whose source or sink value be equal to zero
5:        MASS FLOW ASSIGNING–PHASE ON THE PATH CONSIDERED
          begin
                 If source value is lower than sink value and, Γ(i)=1 Then
                      STOP {infeasible flow}
                 end
                 if source value is lower than sink value and, Γ(i)>1 Then
                     Assign source value on whole path and update source and sink value
                 end
                 If sink value is bigger than source value and this is bigger than capacity arc Then
                      Assign the maximum capacity on path and let the flow remaining for further paths
                      and update source and sink value
             end
                 If sink value is lower than source value Then
                     Assign the source value on whole path and, and update source and sink value
                 end
                 Γ(i) → Γ(i) – 1
                 If ∃ other path for sink node i Then
                    Take the new path and go to step 5
             end
          end
6:  endWhile
7:  RETURN {feasible flows obtained}
```

Figure 3. Procedure for getting feasible flows.

PROCEDURE-2 {Quest of feasible pressures}
begin

0: Input: Assign a initial suction pressure at node r | rεV and it has not incoming arcs.
Hint:
$p_r = \sqrt{(B(r))^2 R_{rj}}$ where B(r) corresponds at its source value and, Rr_j is the resistance of pipeline (r,j)

1: For ∀(i,j)εG=(N,A) moves on the net starting with node r Do
2: For each outgoing arc selecting first a pipeline arc do
3: If (i,j)εAd then
begin
Assign discharge pressure like following: $p_j = p_i^2 - R_{ij} X_{ij}^2$ where
$R_{ij} = C\frac{(fL)}{d^5}$ since L (longitud), d (inner pipeline), f(friction) are features of pipeline
where $C=KS_gT_s$ and $K=1.33050X10^5$
If $p_j < p_j^L$ Then
Come back until to find a compressor arc or there is not futher incoming arcs
end
end
4: If (i,j)εAc then
begin
Obtain the station compressor domain
Using the lower and upper bound on suction and discharge pressure obtained in the preprocessing technique.
And,
Assign $p_j = p_j^L$
end
5: end For
6: For each incoming arc selecting first a pipeline arc do
7: If (i,j)εAd then
begin
Assign discharge pressure like following: $p_i = p_j^2 + R_{ij} X_{ij}^2$ where
$R_{ij} = C\frac{(fL)}{d^5}$ since L (longitud), d (inner pipeline), f(friction) are features of pipeline
where $C=KS_gT_s$ and $K=1.33050X10^5$
If $p_i < p_i^L$ Then
Go until to find a compressor arc or there is not further outgoing arcs
end
end
8: If (i,j)εAc then
begin
Obtain the station compressor domain
Using the lower and upper bound on suction and discharge pressure obtained in the preprocessing technique.
And,
Assign $p_i = p_i^U$
end
9: end For
10: end For
11: **RETURN** {feasible pressures}

Figure 4. Procedure for getting feasible pressures.