Optimization of Natural Gas Pipeline Transportation using Ant Colony Optimization Algorithm

A. Chebouba[*,], L. Amadeo[*], F. Yalaoui[*], A. Smati[**]**

[*] *ICD, UTT, FRE CNRS 2848, LOSI, 12 rue Marie Curie, BP 2060, 10010 Troyes cedex, France.*
[**] *Laboratory of Reliability of the oil equipments and the materials, Faculty of Hydrocarbon and Chemistry, University of Boumerdes, Algeria*
chebouba@yahoo.fr

ABSTRACT: In this paper, an Ant Colony Optimization Algorithm (ACO) is proposed for operations of steady flow gas pipeline. The system is composed of compressing stations linked by pipe legs. The decisions variables are chosen to be the operating turbo compressor number and the discharge pressure for each compressing station. The objective function is the power consumed in the system by these stations. Until now, essentially Gradient based procedures and dynamic programming have been applied for solving this no convex problem. The main original contribution proposed, in this paper, is that we use an ant colony optimization algorithm for this problem. This method was applied to real life situation. The results are compared with those obtained by employing dynamic programming method. We obtain that compared with those obtained by employing dynamic method. We obtain that the ACO is an interesting way for the gas pipeline operation optimization.

RESUME : Dans ce papier, nous proposons une formulation développée à partir des algorithmes d'optimisation de colonies de fourmis (ACO) pour la détermination des régimes de fonctionnement d'un gazoduc. L'optimisation des régimes de fonctionnement consiste à minimiser la consommation de gaz carburant des stations de compression. Les variables de décision sont les pressions de refoulement de ces stations et le nombre de groupes turbocompresseur à mettre en service dans chacune de ces stations. Cette formulation a été appliquée au gazoduc Hassi R'mell- Arzew. Les résultats sont comparés avec ceux obtenus en employant la méthode de la programmation dynamique. A partir de ces résultats, on peut affirmer que la technique des ACO est une alternative intéressante à plus d'un titre pour l'optimisation des régimes de fonctionnement d'un gazoduc.

KEYWORDS: Gas transport, Optimization, Ant Colony Algorithm.

MOTS CLES: Transport du gaz, Optimisation, Algorithme des colonies de fourmis.

1. Introduction

The gas pipelines are wide complex systems in length (several hundred kilometers, and even of the thousands) intended for the transport of natural gas by pipe. There are three different kinds of topologies: linear (i.e., gun barrel), tree (i.e., branched), and cyclic. A gas pipeline is composed of compressing stations (CS) intended to provide the energy of pressure necessary to transport gas via a pipeline. A number of turbocompressors located in parallel are the principal equipments of the CS. A part of the gas crossing through the station is used as fuel gas for the turbocompressors (TC).

In natural gas pipeline operations, the station operator is responsible for making two important decisions: increase or decrease compression in the pipelines, and start-up or shut down of turbocompressor units. Incorrect decisions made by the operator increases energy cost or may cause customer dissatisfaction. The main objective of this study is to provide a decision aid tool that assists operators to make the most appropriate decision within a short time.

The objective function is the power consumed in the system by these stations. The decisions variables are chosen to be the operating turbocompressor number and the discharge pressure for each compressing station.

Several methods were developed and none of them has considered all aspects of the problem. The majority of them are based on the dynamic programming (DP) [16, 18, 19, 15, 3, 4] or gradient search techniques [14, 20].

The use of metaheuristic, for gas pipeline fuel consumption minimization problem (GPFCMP), is generally limited until now to the genetic algorithms [11]. In Goldberg’s work, very original by the way, not all the constraints of the problem (operation range of the compressors) have been considered.

The principal advantages of DP are that a global optimum is guaranteed and that the nonlinearity can be easily treated. The disadvantages of DP are that its application is practically limited to the simple network topologies (linear or ramified) and that computation increases in an exponential way with problem dimension. The advantages of GRG method consist of that dimension does not be a problem and, it could be applied to cyclic network schemes. However, as GRG method is based on a search gradient method, there is no guarantee to find a global optimum. In fact, with discrete decision variables, it can fix with the local optimum.

The main original contribution proposed in this paper, is that we use, for the first time, the ant colony optimization algorithm for the GPFCMP. The results obtained with the suggested approach are excellent with a strong computing time saving compared to those obtained with the DP technique. This will enable us to design a fast, effective and robust decision aid tool based on the suggested method. This tool will assist operators to make the most appropriate decision within a short time.

The results reported in this work have been applied to the gas transportation pipe "Hassi R'mell-Arzew "of Algerian network.

This method was inspired by ants behaviour studies [6] (Deneubourg *et al.*). The ant algorithm is a new evolutionary optimization method first proposed by Dorigo *et al.* [7] to solve different combinatorial optimization problems like the travelling salesman problem and the quadratic assignment problem. Dorigo and Di Caro [8] introduced the ant colony metaheuristic framework. This enables ACO to be applied to other engineering problems. Abbaspour *et al.* [1] used ACO algorithms to estimate hydraulic parameters of unsaturated soil. Maier *et al.* [12] developed ACO algorithms to find a near global optimal solution to a water distribution system. However, no application of ACO was carried out for gas pipeline operation optimization.

The paper is organized as follows. In chapter 2, the description, the formulation and the assumptions are described. The proposed methodology is fully described in chapter 3. An extensive computational evaluation of the metaheuristic, including comparison with dynamic programming technique, is presented in chapter 4. Finally, we conclude this work in chapter 5.

2. The Problem

2.1 *Description*

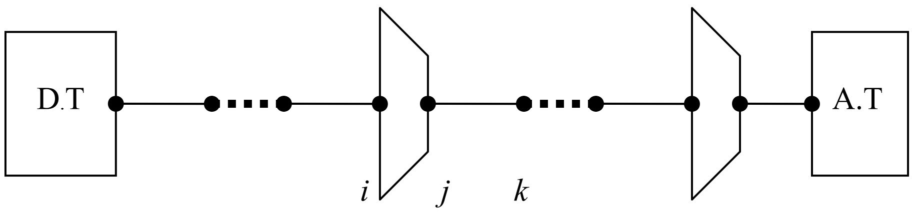


Figure 1. *The transportation system*

D.T.: Departure terminal (Gas gathering and treatment centre);

A.T.: Arrival terminal (Consumer);

$(i,j) \in Ac$;

$(j,k) \in Ap$;

With: Ap: Set of pipe leg arcs.

Ac: Set of compressing station arcs.

V: Set of all nodes.

A: Set of all arcs.

2.2. ***Formulation***

Objective function:

$$\min \sum_{(i,j)\in A_C} \left(x_{ij} \frac{Z_i R T_i}{\omega} \left[\left(\frac{p_j}{p_i} \right)^{\omega} - 1 \right] \right) / \mu_{ij} \qquad [1]$$

Subject to

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

$$P_i^l < p_i < P_i^u, \qquad i \in V \qquad [3]$$

$$p_j \geq 0 \qquad (i,j) \in Ac \qquad [4]$$

$$(x_{ij}/n_{ij},\, p_i,\, p_j) \in D_{ij} \qquad n_{i,j} \in \{0,1,2,\ldots,N_{ij}\},\ (i,j) \in Ac \qquad [5]$$

With: p_i: Pressure at node i.

p_j: Pressure at node j.

P_i^l, P_i^u: Pressure limits at node i; l= lower limit, u= upper limit.

V: Set of all nodes.

R_{ij}: Resistance of pipe leg (i, j).

x_{ij}: Mass flow rate through the compressor station (i, j).

n_{ij} : Operating turbocompressors number in station (i, j).

N_{ij}: Number of available turbocompressors in the station (i, j).

Z_i: The gas compressibility factor at suction conditions of station (i,j).

R: The gas constant.

T_i: Suction gas temperature in station (i, j).

μ_{ij}: Turbocompressor adiabatic efficiency in station (i, j).

$\omega = \dfrac{\gamma - 1}{\gamma}$ Where γ is the gas specific heat ratio.

D_{ij}: Feasible operating domain for a single turbocompressor unit in compressing station *(i, j)*.

Equation (1) is the total power consumed by all the pipeline compressor stations.

Equation (2) defines the gas flow dynamics in each pipe leg *(i, j)*.

Equation (3) bounds the pressure in the pipeline.

Equation (4) defines the pressure as nonnegative variable.

Equation (5) represents the feasible operating domain for a single turbocompressor unit.

The compressor stations are constituted of several identical turbocompressors, built in parallel, which could be stopped or started. The operation range of a turbocompressor in compressing station *(i,j)* as a function of the variables q_{ij} (flow through the turbocompressor unit), p_i (suction pressure) and p_j (discharge pressure) is given by the following equations .

$$\frac{h_{ij}}{s_{ij}^2} = A_H + B_H\left(\frac{q_{ij}}{s_{ij}}\right) + C_H\left(\frac{q_{ij}}{s_{ij}}\right)^2 + D_H\left(\frac{q_{ij}}{s_{ij}}\right)^3 \quad [6]$$

$$\mu_{ij} = (C_E\left(\frac{q_{ij}}{s_{ij}}\right)^2 + B_E\left(\frac{q_{ij}}{s_{ij}}\right) + A_E)/100 \quad [7]$$

$$S_{min} < s_{ij} < S_{max} \quad [8]$$

$$Surge < q_{ij}/s_{ij} < Stonewall \quad [9]$$

Where A_H, B_H, C_H, D_H, A_E, B_E and C_E are constants which depend on the compressor unit and are typically estimated by applying the least squares method to a set of collected data of the quantities q_{ij}, s_{ij}, h_{ij} et μ_{ij} [17]. *Surge* is lower bound of q_{ij}/s_{ij} and *Stonewall* is upper bound of q_{ij}/s_{ij}.

The relationships between (h_{ij}, q_{ij}) and (x_{ij}, p_i,p_j) are the following:

$$h_{ij} = \frac{Z_i RT_i}{\omega}\left[\left(\frac{p_j}{p_i}\right)^{\omega} - 1\right] \quad [10]$$

$$q_{ij} = Z_i RT_i \frac{x_{ij}}{p_i n_{ij}} \quad [11]$$

Where:

S_{min}: Turbocompressor minimum speed.

S_{max}: Turbocompressor maximum speed.

q_{ij}: Turbocompressor inlet volumetric flow rate in station (i, j)..

s_{ij}: Turbocompressor speed in station (i, j).

h_{ij}: Turbocompressor adiabatic head in station (i, j).

3. Ant colony optimization algorithm

3.1 *Introduction*

The graph G(D, L, C) [21] of the GPFCMP can be represented as a set of nodes $D = \{1,2,\ldots,n+1\}$. Each node $i \leq n$ (n: number of compressor stations) is linked to the next via a set of edges $\zeta(i,S_i^{'}) = \{l(i,j,S_i^{'}) : j = 1,2,\ldots,NO_i\}$, where $l(i,j,S_i^{'})$ is the jth edge(station discharge pressure) connecting node i to node i+1. NO_i is the number of edges connecting node i to node i+1 with the preceding semi-constructed tour $S_i^{'}$ and the set of all edges is $L=\{s: s \in \cup_{i=1}^{n} \zeta(i, S_i^{'})\}$. A feasible tour through this graph is then an element of the solution space $\mathbf{S} = \{S: S = \{s(1,S_1^{'}), \ldots, s(n,S_n^{'})\}, s(i,S_i^{'}) \in \zeta(i,S_i^{'}), i=1,\ldots,n\}$. $C = \{c(i,j,S_i^{'})\}$ is the set of costs associated with edge $l(i,j,S_i^{'})$.

In what follows, for the clearness of the text, we omit to add the preceding semi-constructed tour $S^{'}$ in the formulas.

A set of finite constraints $\Omega(D,L)$ may be assigned over the elements of D and L.

The m ants are placed at the starting node. Ants build a solution to solve GPFCMP, while moving from a node to another one to all visit them.

During an iteration t, each ant k carries out a tour $T^k(t)$, and during this tour, the choice of edge $l_{i,j}$ connecting node i to node i+1, depends on the following criteria:

1. The inverse of cost c_{ij}, called visibility η_{ij} (η_{ij}=1/ c_{ij}). This heuristic value is calculated once at the start of the algorithm and is not changed during the computation.

2. The concentration of pheromone $\tau_{ij}(t)$ on edge l_{ij} at iteration t. The pheromone trail takes into account the ant's current history-performance. The amount of pheromone trail $\tau_{ij}(t)$ associated to edge l_{ij} is indented to represent the learned desirability of choosing jth edge at node i. The pheromone trail information is changed during problem solution to reflect the experience acquired by ants during problem solving.

3.2. *Ant Colony System (ACS)*

ACS algorithm was introduced to improve the performances of the basic algorithm [9] on big size problems, the modifications are as follows [10]:

ACS introduces a rule of transition depending on a parameter q_0 ($0 \le q_0 \le 1$), which determines the relative importance of exploitation versus exploration : every time an ant at node i selects edge l_{ij} according to the following transition rule:

$$J = \begin{cases} argmax_{u \in J_i^k} \left[(\tau_{iu}(t))^{\alpha} (\eta_{iu})^{\beta} \right], & \text{if } q \le q_0 \,; \\ J, & \text{otherwise;} \end{cases} \quad [12]$$

Where q is a random variable uniformly distributed over [0,1] and $J \in J_i^k$ a random value selected according to the probability

$$p_{iJ}^k(t) = \begin{cases} \dfrac{[\tau_{iJ}(t)]^{\alpha} [\eta_{iJ}]^{\beta}}{\sum_{u \in J_i^k} [\tau_{iu}(t)]^{\alpha} [\eta_{iu}]^{\beta}}, & \text{for } J \in J_i^k \,; \\ 0, & \text{for } J \notin J_i^k \,; \end{cases} \quad [13]$$

The parameters α=1 [9] and $\beta \in$[0, 9.5] control the relative importance of the pheromone trail and heuristic value referred to as pheromone and heuristic sensitivity parameters, respectively.

–The pheromone trail is changed both locally and globally.

-Local updating : Every time an edge $l_{i,j}$ is chosen by an ant, the amount of pheromone will change by applying the local trail updating formula :

$$\tau_{ij}(t) = (1-\rho)\tau_{ij}(t) + \rho\, \tau_0 \qquad [14]$$

Where τ_0 is the initial pheromone value, ρ evaporation rate.

-Global updating: Upon completion of a tour by all ants in the colony, the global trail updating is done as follows:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\, \Delta\tau_{ij}(t) \qquad \Delta\tau_{ij}(t) = \frac{1}{L^+} \qquad [15]$$

Where edge $l_{i,j}$ belongs to the best tour within the past total iteration, and L^+ value of the objective function for the ant with the best performance within the past total iteration.

The ACS algorithm works in this application as follows.

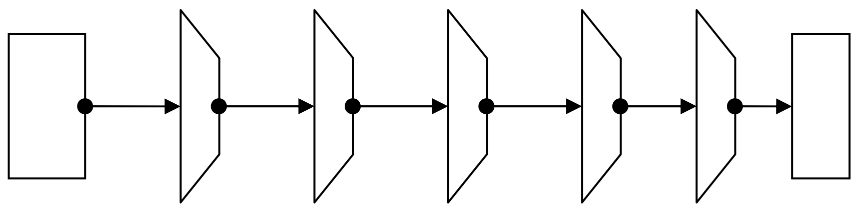
1. *m* ants are initially positioned at the starting node (Departure terminal) in the same edge $l_{i,j}$ (Departure terminal discharge pressure);
2. each ant builds a tour by repeatedly applying the state transition rule;
3. an ant, while constructing its tour, changes the amount of pheromone on the visited edges by applying the local updating rule;
4. seek the best tour using the solution process, in which ants are guided in building their tours by both heuristic information and by pheromone information. An edge with a high amount of pheromone is a very desirable choice;
5. once all ants have terminated their tour, only one ant (the best so far ant) is allowed to add pheromone after each iteration;
6. end conditions (maximum number of iterations or stagnation situation);

4. Description and results

4.1. *Description*

The gas pipeline considered in our calculations, in the first part of this study, is that of “Hassi R' mell-Arzew”. In the second part of this work, we consider general cases having the same principal data of Hassi R'mell-Arzew gas pipeline. However, the compressor station number and the turbocompressor number are different. A name Csx-Nby represents an instance with x compressor station and y turbocompressor.

Hassi R'mell-Arzew gas pipeline is composed of one source, one demand and six pipe-legs connected in series by five compressor stations. These stations are constituted of three identical turbocompressors, built in parallel. A schematic illustration of this pipeline is provided in Fig. 2.



Hassi R’mell Station 1 Station 2 Station 3 Station 4 Station 5
Arzew

Figure 2. Gas pipeline Hassi R’mell-Arzew

Hassi R’mell: Gas gathering and treatment centre.

Arzew: Liquefied natural gas plant.

4.2. *Tests*

The computational tests was developed on a DELL biprocessor workstation with 1 Giga RAM and 440 Mhz. The algorithm is coded using matlab 7.

As with any metaheuristics, many parameters need to be set to have a good performance of ACO algorithm. The model performance was tested against variations of ρ, β, q_0, *m* (ant number) and *t_max* (number of iterations). To have an idea on the best possible values of these parameters, a feasible range for each parameter was first defined. With $\beta \in \{1, 5, 6, 7, 8, 8.5, 9, 9.5\}$, $\rho \in \{0.05, 0.075, 0.1, 0.25, 0.5, 0.75, 0.99\}$ and $q_0 \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ objective function values were observed to select the best combination of these parameters.

A value of 0.1 for the pheromone evaporation rate, one of 8 for β and one of 0.5 for q_0 seem to be the best choice for our problem.

4.3. *Hassi R'mell-Arzew Gas pipeline case*

This gas pipeline is a Cs5-Nb3 instance. We have compared the solution obtained by proposed method (ACO) with that of dynamic programming [4] for different flow rates (table 1). The two last columns of this table show the relative error (*RE*) and the computing time saving in percent (*CTSP*) of ACO over DP, given by

$$RE = \frac{\left| Objective\,value_{ACO} - Objective\,value_{DP} \right|}{Objective\,value_{DP}} * 100$$

$$CTSP = \frac{\left| CPU_{DP} - CPU_{ACO} \right|}{CPU_{DP}} * 100$$

Table 1: *Computing time and objective value for Hassi R'mell-Arzew Gas pipeline*

Flow rate	ACO		DP		*CTSP*	*RE*
	CPU time (s)	Objective value	CPU time (s)	Objective value		
950000	209	17182	1727	17178	88	0,02
1000000	203	23602	1170	23591	83	0,05
1050000	199	28426	1446	28409	86	0,06
1100000	195	34257	1544	34254	87	0,009
1150000	185	41811	1766	41803	89	0.02

From table 1, we can see that the ACO is still almost good as the DP and the computing time saving in percent of ACO over DP is bigger than 83. It becomes 6-9 times faster. We notice also, for some flow rates (1150000 kg/h) (table 2), that the results of calculation of the discharge pressures are in agreement with Batey's principle [2].

This principle very known for gas pipeline engineers can be expressed as follows: *all the compressor station has to work in the most raised possible pressure (in our case 72 bars) except the last one who has to develop a sufficient just load so that pressure in the arrival is equal to the acceptable minimal pressure.*

In table 2, we present an optimal policy obtained by the proposed method for a flow rate of 1150000 kg/h.

Table 2: *Optimal policy for a flow rate of 1150000 kg/h.*

stations	**Variables**	**Flow rate (kg/h)**
		1150000
Station 1	Discharge pressure (bars)	72
	Operating turbocompressors	3
Station 2	Discharge pressure (bars)	72
	Operating turbocompressors	3
Station 3	Discharge pressure (bars)	72
	Operating turbocompressors	3
Station 4	Discharge pressure (bars)	72
	Operating turbocompressors	3
Station 5	Discharge pressure (bars)	64.35
	Operating turbocompressors	3

4.4. *General cases*

To test the algorithm performance in general cases, we consider different problem sizes. For this, we analyze, for different flow rates, the influence of both compressor station number and turbocompressor number on the performance of ACO algorithm.

For these various instances, computing times and objective values were calculated. Tables 3, 4 and 5 show a comparison between DP and ACO. These tables share the same format as that of table 1.

As we can see, from table 3, the relative error of ACO over DP is less than 0.22 %. Moreover, the computing time saving in percent of ACO over DP is bigger than 93. The ACO is 14-27 times faster.

Table 3: *Computing time and objective function for a Cs11-Nb6 instance*

Flow rate	ACO		DP		*CTSP*	*RE*
	CPU time (s)	Objective value	CPU time (s)	Objective value		
950000	721	49217	10332	49114	93	0,21
1000000	509	61641	11533	61534	96	0,17
1050000	477	73413	11779	73252	96	0,22
1100000	445	87028	11335	86942	96	0,10
1150000	418	103739	11150	103600	96	0.13

From table 4, we first observe that relative error of ACO over DP is less than 0.41 %. On the other hand, we also observe that computing time saving in percent of ACO over DP is bigger than 96. In fact, the ACO becomes 23-42 times faster.

Table 4: *Computing time and objective function for a Cs17-Nb9 instance*

Flow rate	ACO		DP		*CTSP*	*RE*
	CPU time (s)	Objective value	CPU time (s)	Objective value		
950000	1251	81152	28262	81050	96	0,13
1000000	1114	99882	30405	99477	96	0,41
1050000	1069	118311	29886	118122	96	0,16
1100000	977	140230	28560	139705	97	0,38
1150000	649	165888	27522	165397	98	0.30

As can be seen from table 5, the relative error of ACO over DP is less than 0.66 %. We can observe also that the computing time saving in percent of ACO over DP is bigger than 97. The ACO becomes 30-57 times faster.

Table 5: *Computing time and objective function for a Cs23-Nb12 instance*

Flow rate	ACO		DP		*CTSP*	*RE*
	CPU time (s)	Objective value	CPU time (s)	Objective value		
950000	1869	113390	55168	112985	97	0,36
1000000	1525	137913	57628	137419	97	0,36
1050000	1536	164062	56332	162993	97	0,66
1100000	1405	193297	53409	192469	97	0,43
1150000	895	228550	50786	227194	98	0,60

Finally, as can be seen from these tables, ACO algorithm gives quasi-optimal solutions in less computing time for all flow rates and for different problem sizes. In some instances, ACO becomes 57 times faster. This shows the effectiveness of the proposed method.

5. Conclusions

In this paper, we use a relevant technique to minimize fuel consumption of gas pipeline. An algorithm based on ant colony metaheuristic was very performing compared to dynamic programming technique. In fact, with the suggested method, we obtain excellent results with a strong computing time saving. This will enable us to design a fast, effective and robust decision aid tool based on the suggested method. This tool will assist operators to make the most appropriate decision within a short time.

A careful sensibility analysis is required for parameters involved in the algorithm (ρ, β and q_0). In this work, some variations of these parameters were tested and the values reported are those that gave us better results.

Finally, obtained results encourage us to study more complex structures (cyclic network topology), non-stationary problem and combinatorial aspects (non-identical turbocompressors…).

References

[1] Abbaspour K. C., Schulin R. and Van Genuchten M., T. "Estimating unsaturated soil hydraulic parameters using ant colony optimization", *Advanced Water Resource*, 2001, 24(8): 827-933.

[2] Batey E. H, Courts H. R. and Hannah K. W, "Dynamic approach to gas pipeline analysis", *Oil & Gas Journal*, 1961, 59, 65-78.

[3] Carter R., "Pipeline optimization: Dynamic programming after 30 Years", *30 th Annual Meeting Pipeline Simulation Interest Group (PSIG)*, 28-30 Oct., Denver, Colorado1998.

[4] Chebouba A. and Smati A., « Optimisation d'un pipeline de transport de gaz naturel par la programmation dynamique avec choix automatique des compresseurs. Conception et production intégrées Meknes », 2003 ; 39 : 1-14.

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

[6] Deneubourg J.L., Pasteels J.M. and Verhaeghe J.C., "Probabilistic behaviour in ants: A strategy of errors", *Journal of Theoritical biology*, 1983; 105: 259-272.

[7] Dorigo M., Maniezo V. and Colomi A., "The ant system: optimization by a colony of cooperating ants", *IEEE Trans. Syst. Man. Cybern,* 1996; 26: 29-42.

[8] Dorigo M. and Di Caro G., *The ant colony optimization metaheuristic. New ideas in optimization,* D. Corne , M. Dorigo, and F. Glover, eds., McGraw-Hill, London, 1999. p. 11-32.

[9] Dorigo M. and Stützle T., *Ant colony optimization,* The MIT press, 2004.

[10]Dréo J., Pétrowski A., Siarry P. and Taillard E. *Métaheuristiques pour l'optimisation difficile,* Eyrolles, 2003.

[11]Goldberg D. E. Computer-aided gas pipeline operation using genetic algorithms and rule learning. Doctoral dissertation, university of Michigan, 1983.

[12]Maier H. R., Simpson A. R., Zecchin A. C., Foong W.K., Phang K. Y., Seah H.Y. and Tan C. L., "Ant colony optimization for design of water distribution systems" 2003, *Journal of water resources planning and management*, ASCE; 129(3): 200-209.

[13] Martin A., Möller M. and Moritz S., "Mixed integer models for the stationary case of gas network optimization", *Mathematical programming,* 2006; 105: 563-582.

[14] Percell P. B. and Ryan M. J. "Steady state optimization of gas pipeline network operation", *Proceedings of the 19th PSIG Annual Meeting*, Tulsa, October 1987.

[15] Peretti A. and Toth P.," Optimization of a pipeline for the natural gas transport", *European journal of operational research*, 1982; 11: 247-254.

[16] Ríos-Mercado R. Z., Kim S. and Boyd E. A., "Efficient operation of natural gas transmission systems: A network-based heuristic for cyclic structures", *Computers & Operations Research*, 2006; 33: 2323-2351.

[17]Walton T. A., "PC program helps monitor efficiency by describing pump, compressor impeller curves", *Oil & Gas Journal*, 1986; Sept. 1: 55-58.

[18] Wong P. J. and Larson R. E., "Optimization of natural gas pipeline system via dynamic programming". *IEEE Transactions on Automatic Control*, 1968; 5(AC-13): 475-481.

[19] Wong P. J. and Larson R. E., "Optimization of tree structured natural gas transmission networks", *Journal of Mathematical Analysis and Applications*, 1968; 24(3): 613-626.

[20] Wu S., Rios-Mercado R. Z., Byod E. A. and Scott L. R, "Model Relaxation for the Fuel Cost Minimization of steady state Gas Pipeline Networks", *Mathematical and Computer Modeling*, 2000; 31(2-3): 197-220.

[21] Zecchin A. C., Simpson A. R., Maier H. R., Leonard M., Roberts A. J. and Berrisford M. J., "Application of two ant colony optimisation algorithms to water distribution system optimisation", *Mathematical and computer modelling,* 2006; 44(5-6):451-468.