Computers & Operations Research 36 (2009) 1916 – 1923

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/cor

Optimization of natural gas pipeline transportation using ant colony optimization

A. Chebouba[a,b,*], F. Yalaoui[a], A. Smati[b], L. Amodeo[a], K. Younsi[b], A. Tairi[c]

[a]ICD, UTT, FRE CNRS 2848, LOSI, 12 rue Marie Curie, BP 2060, 10010 Troyes Cedex, France
[b]Laboratory of Reliability of The Oil Equipments and The Materials, Faculty of Hydrocarbon and Chemistry, University of Boumerdès, Algeria
[c]Faculty of Hydrocarbon and Chemistry, University of Boumerdès, Algeria

ARTICLE INFO

Available online 12 June 2008

Keywords:
Steady flow
Gas transport
Optimization
Ant colony
Pipeline

ABSTRACT

In this paper, an ant colony optimization (ACO) algorithm is proposed for operations of steady flow gas pipeline. The system is composed of compressing stations linked by pipelegs. The decisions variables are chosen to be the operating turbocompressor 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 ACO 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 the ACO is an interesting way for the gas pipeline operation optimization.

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 (CSs) intended to provide the energy of pressure necessary to transport gas via a pipeline. A number of turbocompressors (TCs) 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 TCs.

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 TC 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 non-linear and non-convex, and all constraints are non-linear. The gas pipeline operations involve combinatorial aspects: compressor stations are constituted of several non-identical TCs, built in parallel, which could be stopped or started. In the stations, one can decide in which combinations several TC units should be run. The problem is time dependent as pipelines function with different flows to adapt at the request of the customers. A system is said to be in steady state when the gas flow values in the system are independent of time. As described, the problem is intractable today [1]. In the present paper, as our system will operate for a relative large amount of time, we can assume that the problem is in steady state. In this study, we focus on gun barrel case composed of CSs constituted of several identical TCs. The objective function is the power consumed in the system by these stations. The decisions variables are chosen to be the operating TC number and the discharge pressure for each CS.

Several methods were developed and none of them have considered all aspects of the problem. The majority of them are based on the dynamic programming (DP) [2–7] or gradient search techniques [8,9].

The work of Batey et al. [10] represents one of the early attempts to develop a rational control policy. Wong and Larson [3] applied the DP technique for the first time to the linear systems in 1968, and then Wong and Larson [4] used it for the ramified networks. Peretti and Toth [5] used a DP formulation, which uses dominance properties and lower bounds. In 1987, Percell and Ryan [8] applied a methodology based on a generalised reduced gradient (GRG) nonlinear optimization technique for non-cyclic structures. The most significant work on cyclic networks is due to Carter [6] who used a non-sequential DP algorithm, but limited to a set of flows. Wu et al. [9] presented two model relaxations, one for the feasible operating domain of compressor and another in the fuel cost function, and derive a lower bounding scheme. Cobos-Zaleta and Ríos-Mercado [11]

* Corresponding author at: ICD, UTT, FRE CNRS 2848, LOSI, 12 rue Marie Curie, BP 2060, 10010 Troyes Cedex, France.
E-mail address: chebouba@yahoo.fr (A. Chebouba).

doi:10.1016/j.cor.2008.06.005

presented a method based on an outer approximation with equality relaxation and augmented penalty algorithm (OA/ER/AP) for solving a MINLP problem. Chebouba and Smati [7] proposed a model based on DP for operating compressor unit choice. Ríos-Mercado et al. [2] recently used a two stage iterative procedure. At the first stage, gas flow variables are fixed and optimal pressure variables are found via dynamical programming. At the second stage, they search a new set of flow variables that improve objective function. Martin et al. [1] applied a technique for a piece-wise linear approximation of the model non-linearities. The use of metaheuristic, for gas pipeline fuel consumption minimization problem (GPFCMP), is generally limited until now to the genetic algorithms [12]. In Goldberg's work, 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 non-linearity 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 (ACO) 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.

ACO was inspired by ants behaviour studies [13]. The ant algorithm is a new evolutionary optimization method first proposed by Dorigo et al. [14] to solve different combinatorial optimization problems like the travelling salesman problem and the quadratic assignment problem. Dorigo and Di Caro [15] introduced the ant colony metaheuristic framework. This enables ACO to be applied to other engineering problems. Abbaspour et al. [16] used ACO algorithms to estimate hydraulic parameters of unsaturated soil. Maier et al. [17] 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 Section 2, the description, the formulation and the assumptions are described. The proposed methodology is fully described in Section 3. An extensive computational evaluation of the metaheuristic, including comparison with DP technique, is presented in Section 4. Finally, we conclude this work in Section 5.

2. The problem

2.1. Description

The transportation system is defined by the set of all nodes V, the set of all arcs A, the set of CS arcs A_C and the set of pipe leg arcs A_P.

The transportation system (Fig. 1) receives at its inlet (D.T.) a quantity of gas at some pressure conditions. For each station (i,j), decisions variables are the discharge pressure (p_j) and number of operating TCS (n_{ij}). It is necessary to take into account several constraints. Some of these are physical and represent the feasible operating domain for compressor station (the TC speed and flow rate). Others are management constraints such as the maximum pressure value inside pipes, the minimum and maximum pressure value at suction and discharge node of each station, the minimum pressure value at the outlet of the line (A.T.), the maximum available number of TCs. The subject of our work is the fuel consumption minimization problem of gas pipeline. The problem can be formulated in the following way: for a given flow at the input of the line with certain pressure conditions, and one at the consumer nodes, with a pre-assigned pressure at the end of the line, what are the optimal decision variables values for each station?

2.2. Formulation

Objective function:

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

s.t.

$$p_i^2 - p_j^2 = R_{ij} x_{ij}^2, \quad (i,j) \in A_P \tag{2}$$

$$P_i^l < p_i < P_i^u, \quad i \in V \tag{3}$$

$$p_j \geqslant 0, \quad (i,j) \in A_C \tag{4}$$

$$(x_{ij}/n_{ij}, p_i, p_j) \in D_{ij}, \quad n_{i,j} \in \{0,1,2,\ldots,N_{ij}\}, \quad (i,j) \in A_C \tag{5}$$

At each arc $(i,j) \in A_C$, p_j and p_i are, respectively, the discharge pressure and the inlet pressure of station (i,j). P_i^l and P_i^u are the pressure limits at node i, and represent respectively lower limit and upper limit. R_{ij} represents the resistance of pipeleg (i,j) ($(i,j) \in A_P$). For each station (i,j), x_{ij}, n_{ij} and N_{ij} represent, respectively, the mass flow rate, the operating TCs number and number of available TCs in the compressor station (i,j). The gas compressibility factor Z_i and gas temperature T_i are defined at suction conditions of station (i,j). μ_{ij} is the TC adiabatic efficiency in station (i,j). The gas constant R and the gas specific heat ratio γ are characteristics of the transported gas. D_{ij} is the feasible operating domain for a single TC unit in CS (i,j).

For measuring total power consumed by all the pipeline compressor stations, we use Eq. (1), the TC adiabatic efficiency μ_{ij} is obtained from Eq. (7). Eq. (2) defines the gas flow dynamics in each pipe leg (i,j). Constraint (3) bounds the pressure in the pipeline. Eq. (4) defines the pressure as non-negative variable. Constraint (5) represents the feasible operating domain for a single TC unit. This equation defines that the operating point of the TC must belong to the feasible operating domain which is bounded by inequalities (8) and (9).

The compressor stations are constituted of several identical TCs, built in parallel, which could be stopped or started. Fig. 2. shows CS representation with two parallel identical TCs.

The operation range of a TC in CS (i,j) as a function of the variables q_{ij} (flow through the TC unit), p_i (suction pressure) and p_j (discharge pressure) is given by the following equations:

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

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

$$S_{\min} < s_{ij} < S_{\max} \tag{8}$$

$$Surge < q_{ij}/s_{ij} < Stonewall \tag{9}$$

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} [18]. *Surge* is lower bound of q_{ij}/s_{ij} and *Stonewall* is upper bound of q_{ij}/s_{ij}.

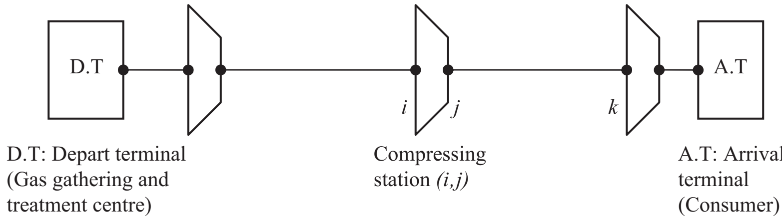

Fig. 1. The transportation system.

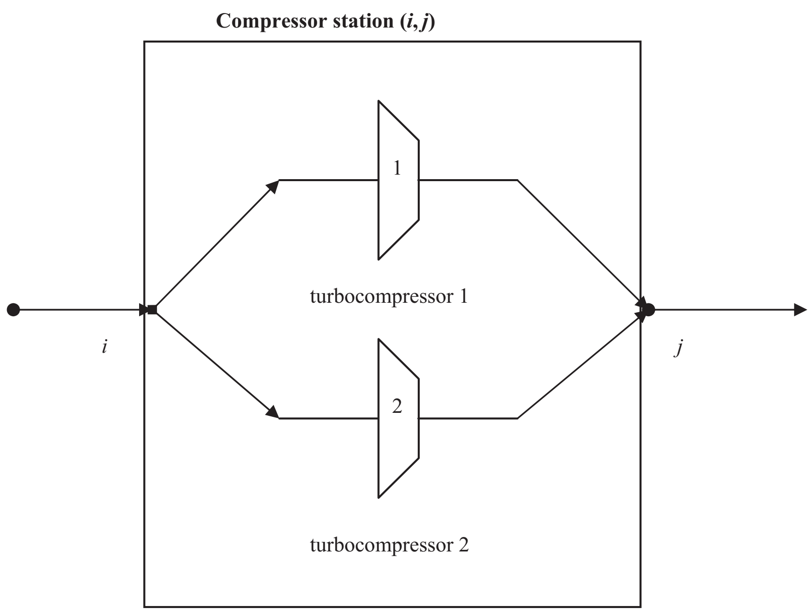

Fig. 2. Schematic of compressor station (i,j).

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

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

$$q_{ij} = Z_i R T_i \frac{x_{ij}}{p_i n_{ij}} \tag{11}$$

The minimum TC speed $S_{\min}$ and the maximum TC speed $S_{\max}$ are the values which bound the TC speed s_{ij} in station (i,j) (Eq. (8)). The TC adiabatic head h_{ij} in station (i,j) is obtained from Eq. (10).

3. ACO algorithm

3.1. Introduction

The graph $G(D,L,C)$ [19] of the GPFCMP can be represented as a set of nodes $D=\{1,2,\ldots,n+1\}$. Each node $i \leqslant n$ (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 $L=\{s : s \in \bigcup_{i=1}^{n} \zeta(i,S'_i)\}$. A feasible tour through this graph is then an element of the solution space $S=\{S : S=\{s(1,S'_1),\ldots,s(n,S'_n)\}, s(i,S'_i) \in \zeta(i,S'_i), i=1,\ldots,n\}$. $C=\{c(i,j,S'_i)\}$ 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 them visit.

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 followings:

1. The inverse of cost c_{ij}, called visibility η_{ij} ($\eta_{ij}=1/c_{ij}$). 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 with 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 [20] on big size problems, the modifications are as follows [21]:

Firstly, ACS introduces a rule of transition depending on a parameter q_0 ($0 \leqslant q_0 \leqslant 1$), 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:

$$j = \begin{cases} argmax_{u \in J_i^k} \lfloor (\tau_{iu}(t))^{\alpha} (\eta_{iu})^{\beta} \rfloor & \text{if } q \leqslant q_0 \\ J & \text{otherwise} \end{cases} \quad (12)$$

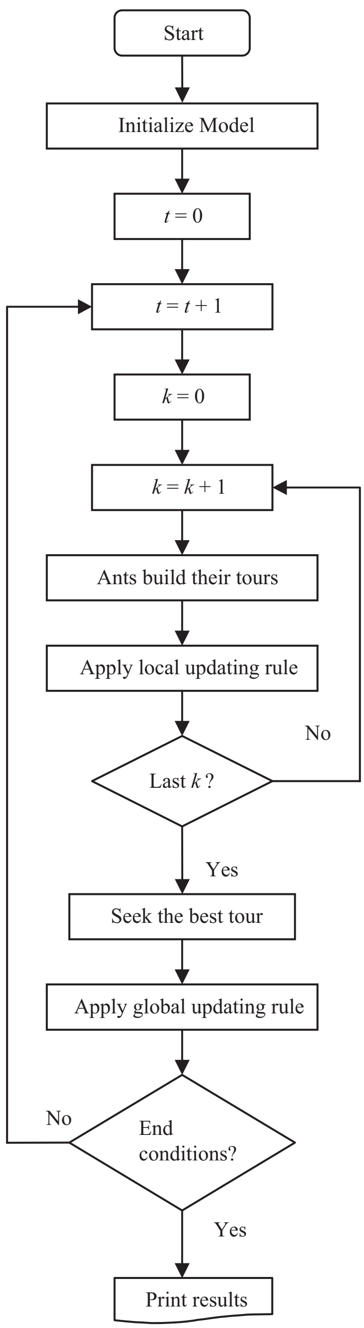

Fig. 3. Steps in ACS algorithm.

Table 1

Pipeleg resistances (R_{ij})	[1.5883 1.4824 1.7154 1.4613 2.1601 2.3296] ∗ 1e8
Gas constant (R)	435 J/(kg K)
Maximum operating pressure (P^u)	72 bars
Minimum suction pressure (P^l)	47 bars
Arzew delivery pressure (P_{TA})	42 bars

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 $\alpha = 1$ [20] and $\beta \geqslant 0$ control the relative importance of the pheromone trail and heuristic value referred to as pheromone and heuristic sensitivity parameters, respectively.

Secondly, in the ACS algorithm, 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:

$$\tau_{ij}(t) = (1 - \rho)\tau_{ij}(t) + \rho\tau_0 \quad (14)$$

  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:

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

  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 main steps in the ACS algorithm are shown in Fig. 3 and include:

1. Ants build their tours as they move from one decision point to the next until all decision points have been covered.
2. The cost of the tour generated is calculated, pheromone is updated locally.
3. After the completion of one iteration (t), pheromone is updated globally.

4. Case study

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 CS number and the TC

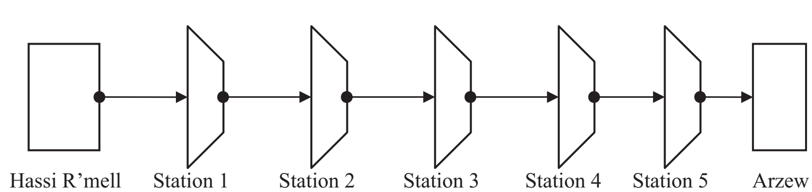

Fig. 4. Gas pipeline Hassi R'mell-Arzew.

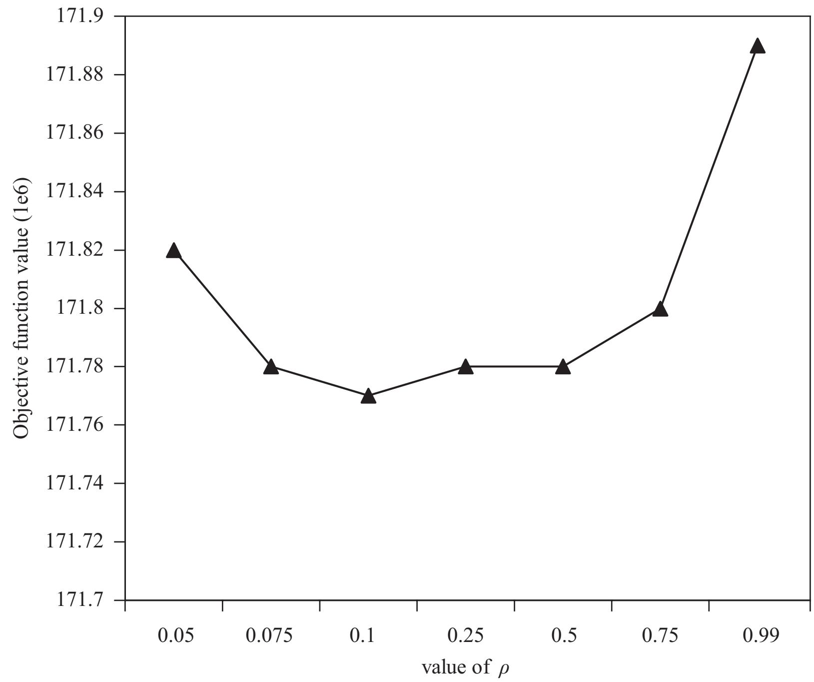

Fig. 5. Behaviour of parameter ρ.

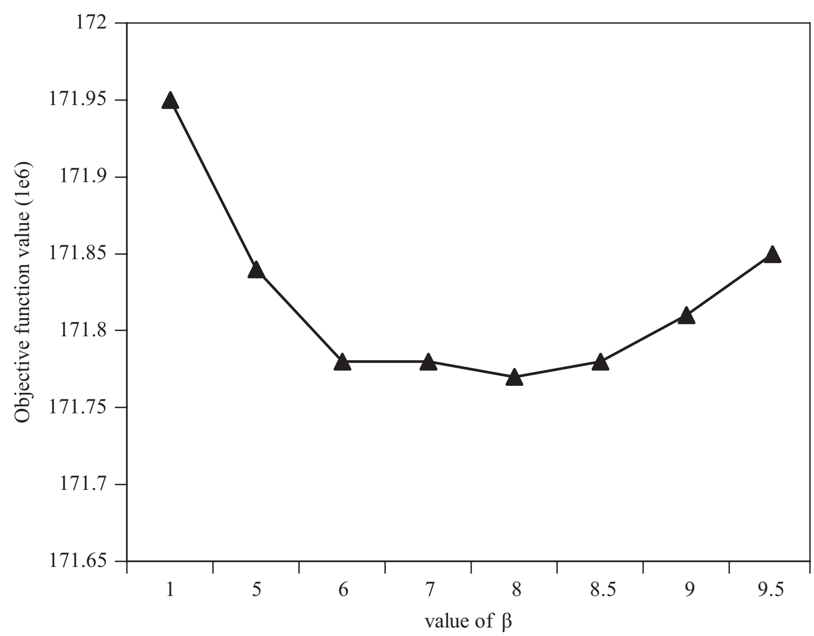

Fig. 6. Behaviour of parameter β.

number are different. A name Cs*x*-Nb*y* represents an instance with *x* CS and *y* TC. For example, an instance of Cs5–Nb3 means that we have five CSs (Cs5) and, in each compressor station, there are three TCs (Nb3).

Hassi R'mell-Arzew gas pipeline is composed of one source, one demand and six pipelegs connected in series by five CSs. These stations are constituted of three identical TCs, built in parallel. A schematic illustration of this pipeline is provided in Fig. 4.

Hassi R'mell: Gas gathering and treatment centre. Arzew: Liquefied natural gas plant.

The principal data for this pipeline are listed in Table 1.

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 $\beta \in \{1, 5, 6, 7, 8, 8.5, 9, 9.5\}$, $\rho \in$

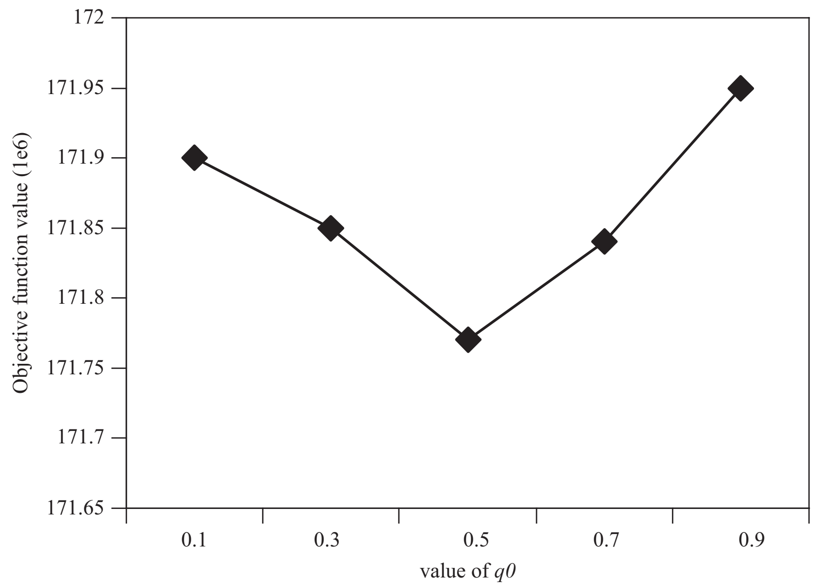

Fig. 7. Behaviour of parameter q_0.

Table 2
Influence of number of iterations and number of ants on objective function.

m		$t_{\_max}$			
		5	**10**	15	20
10	Mean	1.8095	1.7355	1.7207	1.7208
	The best	1.7195	1.7192	1.7188	1.7187
	The worst	1.8699	1.8706	1.7244	1.7247
20	Mean	1.7205	1.7194	1.7197	1.7192
	The best	1.7184	1.7185	1.7182	1.7181
	The worst	1.7245	1.7217	1.7212	1.7210
30	Mean	1.7198	1.7190	1.7190	1.7190
	The best	1.7184	1.7182	1.7182	1.7180
	The worst	1.7218	1.7203	1.7215	1.7213
50	Mean	1.7192	1.7187	1.7184	1.7182
	The best	1.7178	**1.7177**	1.7177	1.7177
	The worst	1.7206	1.7196	1.7190	1.7190

Bold values give the best combination of parameters m and $t_{\_max}$.
$\alpha = 1$, $\beta = 8$, $\rho = 0.1$ and $q_0 = 0.5$.

Table 3
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		
950 000	209	17 182	1727	17 178	88	0.02
1 000 000	203	23 602	1170	23 591	83	0.05
1 050 000	199	28 426	1446	28 409	86	0.06
1 100 000	195	34 257	1544	34 254	87	0.009
1 150 000	185	41 811	1766	41 803	89	0.02

$\{0.05, 0.075, 0.1, 0.25, 0.5, 0.75, 0.99\}$ 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 as shown in Figs. 5–7.

Results reported in Fig. 5 are based on 10 runs (10 restarts) for $\beta = 8$, $\alpha = 1$ and $q_0 = 0.5$.

Results reported in Fig. 6 are based on 10 runs (10 restarts) for $\rho = 0.1$, $\alpha = 1$ and $q_0 = 0.5$.

Results reported in Fig. 7 are based on 10 runs (10 restarts) for $\rho = 0.1$, $\alpha = 1$ and $\beta = 0.8$.

Table 2 presents the objective function values for different number of ants and number of iterations. As expected the quality of the final solution improves as these numbers increase. Results reported in Table 2 are based on 10 runs (10 restarts).

ACO algorithms are initially developed for discrete optimization as for TSP problem. Even, developments were done for specifically continuous process as [22,23]. In some cases and under some considerations we can use the discrete form of the algorithm even if we deal with a continuous problem. This is our case and the case of the algorithm developed in this paper. The emphasis of the paper is to use ACO as a rapid decision aid tool, it is not necessary to use an increment smaller than the value of $\Delta p = 0.01$ bars. As for other algorithm parameters, we test different values in no increasing order of Δp and we adopt the indicated value. This scheme is sufficient for our problem according to industrial practice.

All the following results are obtained with $\alpha = 1$, $\beta = 8$, $\rho = 0.1$, $q_0 = 0.5$, $m = 50$ and $t_{\_max} = 10$.

Fig. 8 shows the effect of discretization size on the objective function value. It is seen that objective function value is decreased for finer discretization.

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 DP [7] for different flow rates (Table 3). 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

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

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

From Table 3, 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 (1 150 000 kg/h) (Table 4), that the results of calculation of the discharge pressures are in agreement with Batey's principle [10].

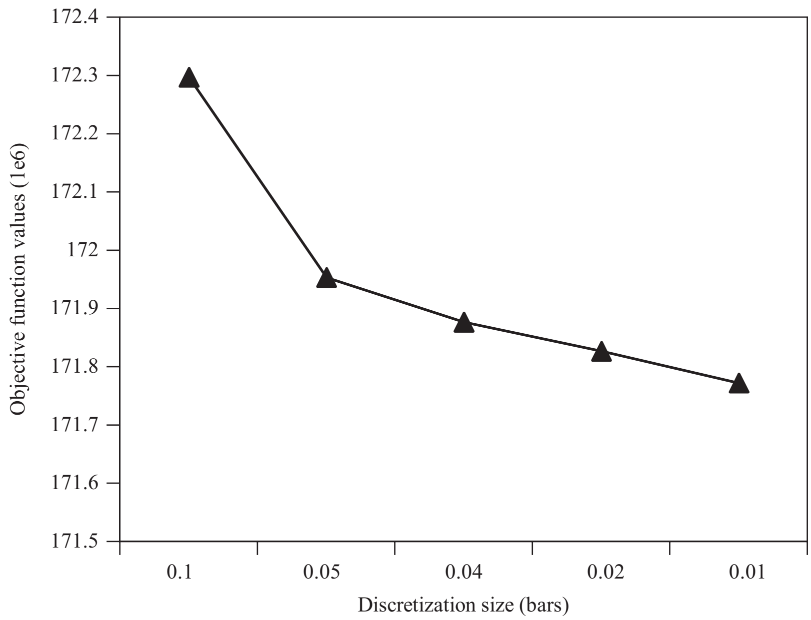

Fig. 8. Influence of discretization size Δp on objective function values. ($\alpha = 1$, $\beta = 8$, $\rho = 0.1$ and $q_0 = 0.5$).

Table 4
Optimal policy for a flow rate of 1 150 000 kg/h.

Stations	Variables	Flow rate (kg/h) 1 150 000
1	Discharge pressure (bars)	72
	Operating turbocompressors	3
2	Discharge pressure (bars)	72
	Operating turbocompressors	3
3	Discharge pressure (bars)	72
	Operating turbocompressors	3
4	Discharge pressure (bars)	72
	Operating turbocompressors	3
5	Discharge pressure (bars)	64.35
	Operating turbocompressors	3

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*. This principle works only for flow rates which are near the nominal flow rate.

In Table 4, we present an optimal policy obtained by the proposed method for a flow rate of 1 150 000 kg/h.

4.4. General cases

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

For these various instances, computing times and objective values were calculated. Tables 5–7 show a comparison between DP and ACO. These tables share the same format as that of Table 3.

As we can see, from Table 5, the relative error of ACO over DP is less than 0.22%.

Moreover, the computing time saving in precent of ACO over DP is bigger than 93. The ACO is 14–27 times faster.

Table 5
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		
950 000	721	49 217	10 332	49 114	93	0.21
1 000 000	509	61 641	11 533	61 534	96	0.17
1 050 000	477	73 413	11 779	73 252	96	0.22
1 100 000	445	87 028	11 335	86 942	96	0.10
1 150 000	418	103 739	11 150	103 600	96	0.13

Table 6
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		
950 000	1251	81 152	28 262	81 050	96	0.13
1 000 000	1114	99 882	30 405	99 477	96	0.41
1 050 000	1069	118 311	29 886	118 122	96	0.16
1 100 000	977	140 230	28 560	139 705	97	0.38
1 150 000	649	165 888	27 522	165 397	98	0.30

Table 7
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		
950 000	1869	113 390	55 168	112 985	97	0.36
1 000 000	1525	137 913	57 628	137 419	97	0.36
1 050 000	1536	164 062	56 332	162 993	97	0.66
1 100 000	1405	193 297	53 409	192 469	97	0.43
1 150 000	895	228 550	50 786	227 194	98	0.60

From Table 6, 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.

As can be seen from Table 7, 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.

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), nonstationary problem and combinatorial aspects (nonidentical turbocompressors).

References

[1] Martin A, Möller M, Moritz S. Mixed integer models for the stationary case of gas network optimization. Mathematical programming 2006;105:563–82.
[2] Ríos-Mercado RZ, Kim S, Boyd EA. Efficient operation of natural gas transmission systems: a network-based heuristic for cyclic structures. Computers & Operations Research 2006;33:2323–51.
[3] Wong PJ, Larson RE. Optimization of natural gas pipeline system via dynamic programming. IEEE Transactions on Automatic Control 1968;5(AC-13):475–81.
[4] Wong PJ, Larson RE. Optimization of tree structured natural gas transmission networks. Journal of Mathematical Analysis and Applications 1968;24(3): 613–26.
[5] Peretti A, Toth P. Optimization of a pipeline for the natural gas transport. European Journal of Operational Research 1982;11:247–54.
[6] Carter, R. Pipeline optimization: dynamic programming after 30 years. In: 30th annual meeting pipeline simulation interest group (PSIG), 28–30 October, Denver, Colorado, 1998.
[7] Chebouba A, 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.
[8] Percell PB, Ryan MJ. Steady state optimization of gas pipeline network operation. In: Proceedings of the 19th PSIG annual meeting, Tulsa, October, 1987.
[9] Wu S, Rios-Mercado RZ, Byod EA, Scott LR. Model relaxation for the fuel cost minimization of steady state gas pipeline networks. Mathematical and Computer Modeling 2000;31(2–3):197–220.
[10] Batey EH, Courts HR, Hannah KW. Dynamic approach to gas pipeline analysis. Oil & Gas Journal 1961;59:65–78.
[11] Cobos-Zaleta D, Ríos-Mercado RZ. 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.
[12] Goldberg DE. Computer-aided gas pipeline operation using genetic algorithms and rule learning. Doctoral dissertation, University of Michigan, 1983.
[13] Deneubourg JL, Pasteels JM, Verhaeghe JC. Probabilistic behaviour in ants: a strategy of errors. Journal of Theoretical biology 1983;105:259–72.
[14] Dorigo M, Maniezo V, Colomi A. The ant system: optimization by a colony of cooperating ants. IEEE Transactions of Systems, Man and Cybernetics 1996;26: 29–42.
[15] Dorigo M, Di Caro G. The ant colony optimization metaheuristic. In: Corne D, Dorigo M, Glover F, editors. New ideas in optimization. London: McGraw-Hill; 1999. p. 11–32.
[16] Abbaspour KC, Schulin R, Van Genuchten MT. Estimating unsaturated soil hydraulic parameters using ant colony optimization. Advanced Water Resource 2001;24(8):827–933.
[17] Maier HR, Simpson AR, Zecchin AC, Foong WK, Phang KY, Seah HY, Tan CL. Ant colony optimization for design of water distribution systems. Journal of Water Resources Planning and Management, ASCE 2003;129(3):200–9.
[18] Walton TA. PC program helps monitor efficiency by describing pump, compressor impeller curves. Oil & Gas Journal 1986;1(September):55–8.
[19] Zecchin AC, Simpson AR, Maier HR, Leonard M, Roberts AJ, Berrisford MJ. Application of two ant colony optimisation algorithms to water distribution system optimisation. Mathematical and Computer Modeling 2006;44(5–6): 451–68.
[20] Dorigo M, Stützle T. Ant colony optimization. Cambridge, MA: MIT Press; 2004.
[21] Dréo J, Pétrowski A, Siarry P, Taillard E. Métaheuristiques pour l'optimisation difficile. Eyrolles; 2003.
[22] Dréo J, Siarry P. Continuous interacting ant colony algorithm based on dense heterarchy. Future Generation Computer Systems 2004;20:841–56.
[23] Socha K, Dorigo M. Ant colony optimization for continuous domains. European Journal of Operational Research 2008;185(3):1155–73.