I&EC research
Industrial & Engineering Chemistry Research

Article

Cite This: Ind. Eng. Chem. Res. 2018, 57, 14307−14314

pubs.acs.org/IECR

Parallel Hyperheuristic Algorithm for the Design of Pipeline Networks

Paola P. Oteiza,[†,‡] Diego A. Rodríguez,[§,∥] and Nélida B. Brignole*[,†,§]

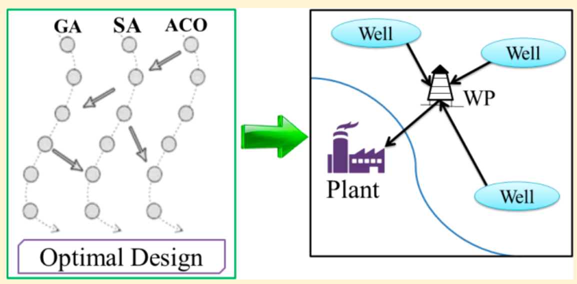
[†]Planta Piloto de Ingeniería Química (PLAPIQUI), Universidad Nacional del Sur (UNS)-CONICET, B8000 Bahía Blanca, Argentina

[‡]Departamento de Ingeniería Química (DIQ), Universidad Nacional del Sur (UNS), B8000 Bahía Blanca, Argentina

[§]Laboratorio de Investigación y Desarrollo en Computación Científica (LIDECC)-Departamento de Ciencias e Ingeniería de la Computación (DCIC), Universidad Nacional del Sur (UNS), B8000 Bahía Blanca, Argentina

[∥]Dpto. de Informática, Facultad de Ciencias Exactas - Universidad Nacional de Salta (UNSa), A4400 Salta, Argentina

ABSTRACT: A hyperheuristic optimization technique to reduce computational times for the design of pipeline networks is presented. The proposed strategy is an A-team approach comprising the guided execution of three metaheuristics: a genetic algorithm, simulated annealing, and an ant colony optimization. Besides, a specialized learning mechanism for information exchange was defined in order to speed up the search process. Moreover, the algorithm was implemented in parallel so as to allow several metaheuristics to run simultaneously, thus achieving a significant reduction of time overhead. In the algorithmic design, realistic scenarios were employed so as to appraise the impact of each agent on optimization efficiency. The cases correspond to real-world offshore infrastructures to be located in the Argentinian marine platform. They were also analyzed to illustrate the validity and suitability of the proposed approach. This optimization technique proved to be competitive since it is able to explore a wide search space fast, yielding satisfactory solutions.



1. INTRODUCTION

The optimization of submarine pipeline networks has proven to be effective to improve operation and gains. Therefore, attention should be devoted to the improvement of computational tools for optimal design of these routes.

Real-world optimization is complex by nature. Several model optimization techniques have been employed to solve problems related to the transport of hydrocarbons from offshore deposits. Van den Heever et al.[1] presented a multiperiod mixed-integer nonlinear programming (MINLP) model, aiming at a long-term design that included economical objectives in detail. To accomplish their goals, they resorted to a specialized heuristic algorithm that determines an offshore hydrocarbon field infrastructure with a production platform, well platforms, and connecting pipelines. Although model complexity caused computational overhead, they reported that larger problems could be solved in a few hours with their proposed method. Moreover, industrial practice was contemplated in their analysis, including considerations about the effect of gas price uncertainty on the solution. In turn, Tarhan et al.[2] presented a MINLP multistage stochastic programming model for the design and planning of infrastructural offshore oil fields under uncertainties. It was pointed out that model size increased exponentially with the number of scenarios and time periods, making their nonconvex MINLP model extremely difficult to solve in full space for real-size problems by means of commercial solvers. Their model yielded good results but exhibited rather long execution times.

In fact, large models for network design problems frequently come up in industrial practice. Ulstein et al.[3] addressed tactical planning, which involved the regulation of production levels of wells, the division of production of oil and gas derivatives, and gas processing and transportation in a network of underwater pipes. In particular, the model was applied to petroleum production in Norway since it was built in collaboration with Statoil, which was an oil company that owned shares in Norwegian petroleum fields. Different cases were analyzed with variations in demand, quality limitations, and system failures. Borraz-Sanchez[4] also addressed examples about pipeline networks connecting wells on the Norwegian continental shelf with the European continent, including the mathematical formulations and difficulties of the corresponding optimization problems. From an operations research perspective, the most important and promising research areas in this field and some real-life applications are reported in Rios-Mercado and Borraz.[5]

The introduction of a nature-inspired approach to these studies has proven to increase profits and efficiency. For example, Baioco et al.[6] described the development of a

Received: June 23, 2018
Revised: September 15, 2018
Accepted: September 27, 2018
Published: September 27, 2018

DOI: 10.1021/acs.iecr.8b02818
Ind. Eng. Chem. Res. 2018, 57, 14307−14314

computational tool for the synthesis and optimization of subsea pipeline routes using evolutionary algorithms. Besides, de Lucena et al.[7] applied genetic algorithms (GA) to improve their analysis for offshore scenarios. In turn, Rocha et al.[8] reported how the optimization of a submarine pipeline route had been enhanced by means of a GA. Later, Baioco et al.[9] successfully focused on multiobjective optimization of subsea pipeline routes in shallow waters.

Recently, in the field of operations research, promising advances were made in the theory and application of metaheuristics aiming at the detection of approximate solutions for optimization problems. In particular, carefully designed hyperheuristics may provide decision-making managers with robust tools to help them in the achievement of economic advantages. Rooted in the area of artificial intelligence, a hyperheuristic[10] is a multilevel search technique that involves a set of solvers (low-level heuristics) and defines priorities in order to select the most useful method at decision points during the process. In general, the choice is a performance evaluation that depends on the search space under exploration.

New hyperheuristic frameworks have been proposed to solve combinatorial optimization problems. For instance, Sabar et al.[11] introduced a gene-expression programming configuration at a higher level. Marshall et al.[12] developed a hyperheuristic approach using grammatical evolution in order to generate heuristics for vehicle routing problems. A complementary review of hyperheuristics can be found in Chen et al.[13] Branke et al.[14] also described some variants to design hyperheuristics, and they remarked that hyperheuristics tend to demand high computing times.

Parallel computing constitutes an effective approach to overcome time delays. Writing and running a parallel program basically depends on the fundamental concept of partitioning the work among the cores. In parallel computing, there are two approaches: either multiple tasks in the program are distributed among the cores or several instances are evaluated simultaneously. Since parallelism has lately become central to the efficient use of resources, exploiting the presence of multiple cores becomes a promising issue in many research fields that will benefit from ever-increasing performance.[15] In particular, Gupta and Grossman[16] have addressed the planning of an offshore oil and gas field infrastructure by applying both approaches. They implemented a parallel decomposition algorithm by using convenient "matrix-partitioning"; they also solved various planning scenarios among the cores. Thanks to this strategy, they achieved satisfactory efficiency values with a time reduction of more than an order of magnitude.

"Matrix-partitioning" is a domain-decomposition method that has frequently been employed to deal with parallel algorithms for systems of equations. The focus of our proposal does not lie on this approach because we do not attempt to solve systems of equations. Instead of working with a descriptive model, our representation of the network topology is based on a graph whose nodes correspond to wells and arcs denote connecting pipes. In contrast, our approach was inspired by Talukdar et al.[17] The organizational framework is an A-team, where the agents run asynchronously and cooperate by reporting their candidate solutions. An agent's task is to perform its own individual search. Each agent is autonomous and completely self-contained, and it was implemented so that it runs sequentially. In other words, the implementation was designed as a multialgorithm problem (MAP), assuming that there is no completely satisfactory algorithm, but several algorithms with an acceptable performance can easily be developed.[18] The multiple-task implementation takes advantage of the best features of three recognized metaheuristics (simulated annealing, genetic algorithm, and ant colony optimization) to guide the search toward a satisfactory solution.

In particular, the design of the connecting pipelines for an offshore oil field was adopted as the main test problem for this paper since the corresponding optimization tasks are time-consuming and particularly difficult to solve.[19] Section 2 describes the real-world problem under study. Next, section 3 outlines the hyperheuristic approach for the optimization of pipeline networks. Section 4 summarizes the test cases, and section 5 contains a brief description of the methodology employed to carry out computational experiments. The algorithmic performance and results are analyzed in section 6. Finally, some conclusions are presented in section 7.

2. PROBLEM STATEMENT

These days computer-aided pipelining is a widespread activity. Network design typically aims at transporting feedstock easily and cheaply so that they can be refined elsewhere into other fuels. An issue of designer's concern is how to locate pipeline networks optimally in short computing times. The hyperheuristic algorithm presented in this paper may constitute a solution to this problem.

In particular, the development planning of an offshore oil and gas field infrastructure is addressed. A typical infrastructure (Figure 1) involves a multiwell site $W = \{W_i, i = 1, ..., z\}$, of z wells with predefined locations and a set of well platforms WP $= \{\mathrm{WP}_k, k = 1, ..., q\}$, whose quantity q and location should be determined. The integer q is an input parameter to specify the upper limit on the number of open platforms.

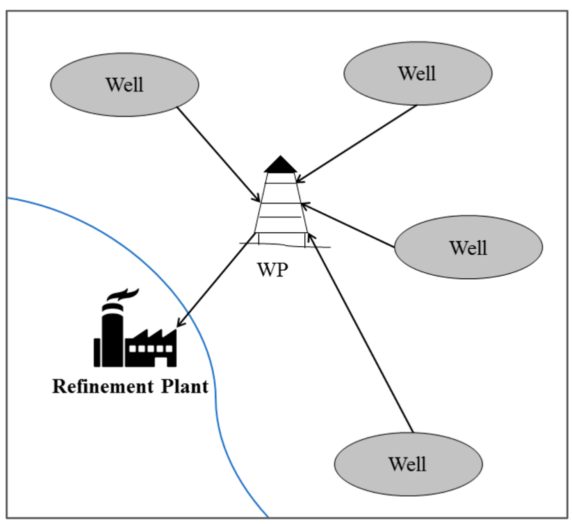


Figure 1. Simplified scheme of the transport network.

The solution strategy to address this problem consists in formulating it as a multiobjective optimization problem (MOOP).

max $\mathbf{F}(\mathbf{x}) = \mathbf{F}\ (f_1\ (\mathbf{x}, ..., f_M(\mathbf{x}))$

Subjected to

DOI: 10.1021/acs.iecr.8b02818
Ind. Eng. Chem. Res. 2018, 57, 14307−14314

$$c_1(\mathbf{x}), \ldots, c_C(\mathbf{x}) \le 0$$
$$d_1(\mathbf{x}), \ldots, d_D(\mathbf{x}) = 0 \tag{1}$$

with $\mathbf{x} \in \mathcal{D}$. The MOOP can be generalized in the form of eq 1, where $\mathcal{D}$ is the decision space, $\mathbf{x}$ is a real N-vector whose elements are the N decision variables, $\mathbf{F}(\mathbf{x})$ is the real-valued multiobjective optimization function, and $f_1(\mathbf{x}), \ldots, f_M(\mathbf{x})$ are the M associated objective functions.

In particular, the objective function (eq 2) was formulated by adopting a conventional weighed aggregation approach.[20] $\mathbf{F}(\mathbf{x})$ is the weighed sum of the objective functions, where the weight w_i for the objective function f_i can be established.

$$\mathbf{F}(\mathbf{x}) = \sum_{i=1}^{M} w_i f_i(\mathbf{x}) \tag{2}$$

Finally, $c_1(\mathbf{x}), \ldots, c_C(\mathbf{x}) \le 0$ and $d_1(\mathbf{x}), \ldots, d_D(\mathbf{x}) = 0$ express the inequality and equality constraints, respectively, all of them imposed on the values of $\mathbf{x}$.

In this network problem, the profit of a network configuration was calculated by its net present value (NPV), the total cost being equal to the revenue minus the expenses (eq 3). For all the metaheuristics, the fitness function $\mathbf{F}(\mathbf{x})$ was formulated by aiming at an NPV maximization to ensure the highest return on the investments over the planning horizon t. The weighed sum of the objective functions (eq 2) was built by including the economic terms in the fitness of individuals as linear aggregated functions.

In eq 3, the expenses of a given configuration are related to the construction and operating costs for each well platform. For the kth platform, its installation cost $\delta_k \ge 0$ is given as its opening cost. The construction costs from the ith well to the jth platform include both pipeline construction $CC_{ij,t}$ and the installation of the corresponding well platform δ_k. The values in $CC_{ij,t}$ are estimated from data about the pipe diameter, flow rate, and the covered distance.[21] The operating costs comprise maintenance $MC_{ij,t}$ and labor LC_t. The transport tariff P_t is included to determine whether the project is economically viable.

$$\begin{aligned} \mathrm{NPV} = {} & \sum_{t=1}^{o} \frac{P_t \times Q_t}{(1+r)^{t+m+n}} - \sum_{t=1}^{n} \sum_{i=1}^{f+q} \sum_{j=1}^{q+p} \frac{CC_{ij,t}}{(1+r)^{t-1}} FP_{ij} \\ & - \sum_{t=1}^{m} \sum_{k=1}^{q} \frac{\delta_k}{(1+r)^{t+n-1}} WP_k \\ & - \sum_{t=1}^{m} \sum_{k=1}^{q} \frac{MC_{k,t}}{(1+r)^{t+n-1}} WP_k - \sum_{t=1}^{o} \frac{LC_t}{(1+r)^{t+m+n}} \end{aligned} \tag{3}$$

The optimization variables are represented as binary numbers to indicate the traversed paths and the installed platforms. In the algorithm, the well platforms are regarded as concentrating nodes, which are located by the optimizer. Then, $WP_k = 1$ if the platform is active; otherwise, $WP_k = 0$. Besides, the integer $FP_{ij} = 1$ indicates that the path running from the ith well to the jth platform is being considered in the itinerary for the configuration; otherwise, $FP_{ij} = 0$.

A network configuration is represented by an occurrence matrix (see an example given in Figure 2), where the last column corresponds to the processing plant (PP).

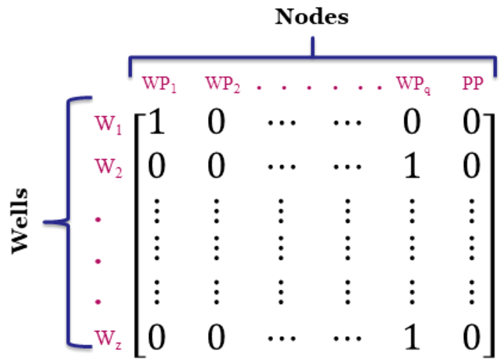


Figure 2. Example of network representation.

3. HYPERHEURISTIC APPROACH

A hyperheuristic is a kind of heuristic that searches a space of low-level heuristics, whereas metaheuristics typically look directly for solutions.[22] The hyperheuristic employs performance information on each low-level heuristic, such as CPU time and solution quality metrics, to choose which one to apply at each stage.

The hyperheuristic algorithm for the design of pipeline networks presented in this work is called a parallel optimizer with hyperheuristics (POWH). It is organized as an A-team,[17] whose agents are autonomous and cooperate by sharing solutions. POWH is based on the master−slave paradigm (Figure 3). The high-level procedure (master) comprises the agent selector and the acceptance criterion. In turn, the low-level strategy is composed of a set of metaheuristics (slaves) that work as optimizing agents. The team comprises the following optimizers: GA (genetic algorithm), SA (simulated annealing), and ACO (ant colony optimization). In GA, the initial population is selected at random, while the best solution designated by the master is always incorporated as an elite solution.

A parallel strategy based on threads was adopted to reduce computational times. In this implementation, several search threads proceed simultaneously, each of them executing a low-level strategy. The queues are managed by means of an index for the optimal assignment of the next task.

As to the allocation of metaheuristics, it is based on their previous behavior. For this purpose, an auxiliary memory is kept with registers of the last executions. Whenever a processor becomes idle, an agent should start working. According to the agent selector, the agent with the highest rank is allocated for the next execution. The choice of agents is controlled by a ranking index (IND_k) whose role is to assess the historic performance of the kth agent. The performance evaluation (IND_k) is defined by eq 4, where eqs 5−8 are the corresponding choice functions. In this strategy, the solution quality (Fe_k), the time consumption (Te_k), the best solution (Oe_k) that could ever be found by the kth agent, and the amount of times (Re_k) the kth procedure has already been applied are considered.

$$IND_k = Fe_k + Te_k + Oe_k + Re_k \tag{4}$$

$$Fe_k = 1 - \left(\frac{Bf_k}{Sf_k}\right) \tag{5}$$

DOI: 10.1021/acs.iecr.8b02818
Ind. Eng. Chem. Res. 2018, 57, 14307−14314

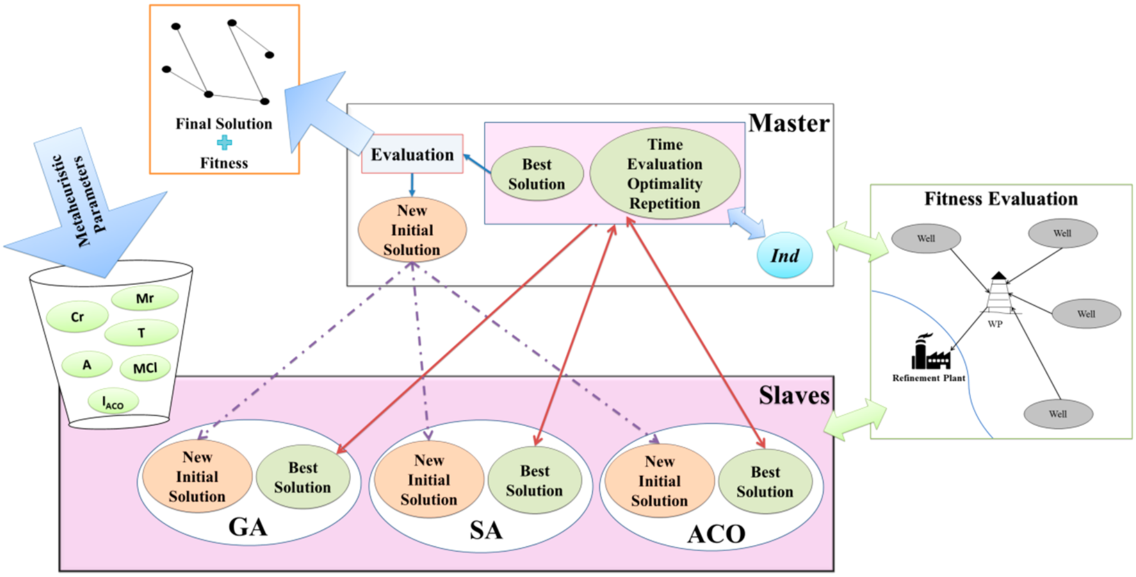


Figure 3. Hyperheuristic design.

$$\mathrm{Te}_k = 1 - \left(\frac{\mathrm{Bt}_k}{\mathrm{St}_k}\right) \tag{6}$$

$$\mathrm{Oe}_k = 1 - \left(\frac{\mathrm{Bf}_k}{\sum_{k=1}^{l} \mathrm{Bf}_k}\right) \tag{7}$$

$$\mathrm{Re}_k = 1 - \left(\frac{e^{R_k}}{\mathrm{Rq}_k}\right) \tag{8}$$

4. TEST CASES

The selected cases for algorithmic testing belong to the Argentinian marine platform. There is a multi-phase flow in the connecting pipelines. It is expected that the facilities and piping connections in the offshore infrastructure will be in operation over t = 30 years.

For the testing procedure, different sizes were considered. Case I (Figure 4) comprised 10 offshore wells and 5 well platforms. For case II (Figure 5), f = 33 dispersed fields were contemplated in the Argentinean sea and the treatment plant was located on the coast of the province of Tierra del Fuego (Argentina).[23] Besides, a grid with q = 18 potential well platforms was considered.

5. METHODOLOGY

For real-life approaches, there are plenty of NP-hard optimization problems that would benefit from quick solutions that may be achieved with the help of parallel programming. To enlarge the scope of the proposed hyperheuristic method, the generalization of the methodology was studied by addressing another challenging problem: a location routing problem (LRP).[24] This computational tool was applied to bus scheduling for the design of Bahia Blanca city with a future perspective of efficient organization.[25]

In view of the promising algorithmic performance, a realistic industrial problem of the optimal design of subsea pipeline networks was chosen to carry out detailed tests. The computational experiments for the design of an adequate A-team consisted in performing the following practical steps:

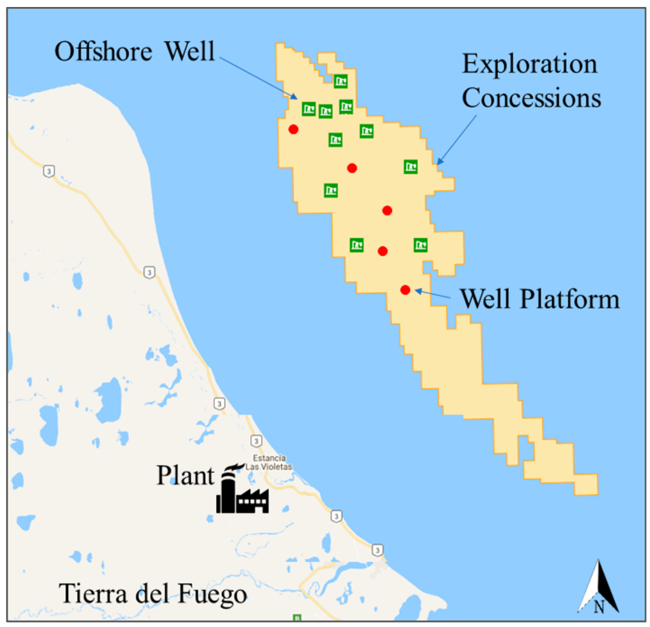


Figure 4. Case I: small instance. Potential locations for subsea pipeline design.

1. to test SA performance for both cases and to find its best parameters
2. to assess the effectivity of the SA–GA cooperation for case II, while tuning GA parameters conveniently
3. to test the hyperheuristics that combine SA, GA, and ACO for case II, while confirming ACO parameters
4. to incorporate parallel programming to the SA–GA–ACO team and to test its performance for case II by comparing parallel with sequential executions

The algorithms were implemented in Java. The executions were always performed on an AMD 8120 eight-core processor with 3.10 GHz and 8 GB of RAM. Thirty executions were adopted for each complete optimization since it is important to

DOI: 10.1021/acs.iecr.8b02818
Ind. Eng. Chem. Res. 2018, 57, 14307−14314

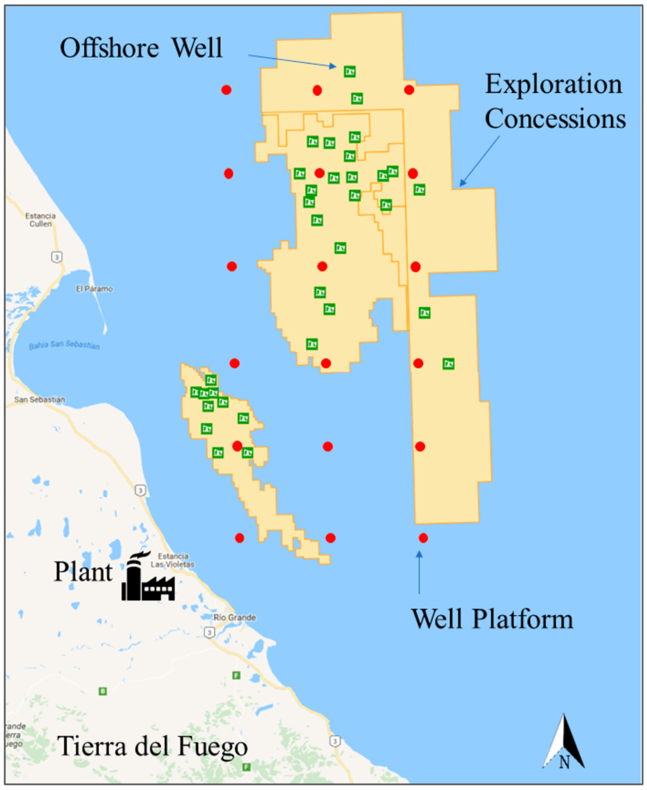


Figure 5. Case II: large instance. Proposed location map for subsea pipeline design.

average the final results over multiple runs because the inner procedures have many random components.

The average fitness values reported in Tables 1–3 were calculated with Internet-released cost data[26,27] for offshore pipeline construction costs in the Argentinean Sea. The construction costs CC_{ijt} are based on both the pipe diameter and the covered distance. For pipelines of o.d. 6–26 in., material costs were calculated with the classic method of cost variation as a function of size.[26] The installation cost of a well platform was regarded as U.S. \$80 000 000.[27] Its maintenance cost (MC_{kt}) was estimated by considering the entire structural value. The operating labor cost was calculated from market conditions, contract structure, and schedule goals based on time and day rate assumptions on work tasks.

Table 1. SA Performance Assessment

	SA			
case	successful hits (%)	average fitness (US\$)	best fitness (US\$)	time (ms)
I (small)	100	9.689×10^7	9.689×10^7	527
II (large)	43	9.441×10^9	9.444×10^9	3279

6. RESULTS

First, SA's performance was assessed (Table 1) by executing 30 independent runs. For case I, the average fitness coincided with the best fitness because the best solution could always be found. In contrast, for case II, the average fitness was much lower than the best one because many unfeasible solutions were found. Only 43% of the independent runs were successful, yielding the best solution satisfactorily.

For case II, Figure 6 shows SA's behavior during 20 iterations in a single execution. The horizontal line shows the best fitness (US\$ 9.444×10^9). This maximum was only reached in iteration 16. It can be observed that SA allowed several times to choose a solution whose fitness value is worse than the current solution. Therefore, the SA algorithm converged relatively slowly toward the final solution.

Next, this drawback was overcome by implementing a cooperation strategy with the help of other metaheuristics. Table 2 shows the improvement of the average fitness when running SA and GA cooperatively. GA contributed to diversify the search by enlarging the search space. Although the exploration was more efficient since 93% of the independent runs could reach the best solution, there was an unfavorable increment in the required time (from 3 279 to 19 345 ms). In order to improve computational times, ACO was finally incorporated into the team because of the remarkable ability of artificial ants to construct solutions guided by the pheromone trails. In this way, a convenient time reduction was achieved thanks to ACO's improved local search procedure. The sequential algorithm with SA, GA, and ACO working in cooperation behaved satisfactorily. In comparison with SA when working alone, SA–GA–ACO's average fitness increased in 10^6 US\$ when the elapsed time almost tripled.

Table 2. Effects of Agent Cooperation in Solving Case II

agents	SA	SA–GA	SA–GA–ACO
average fitness (US\$)	9.441×10^9	9.444×10^9	9.442×10^9
time (ms)	3279	19345	9322

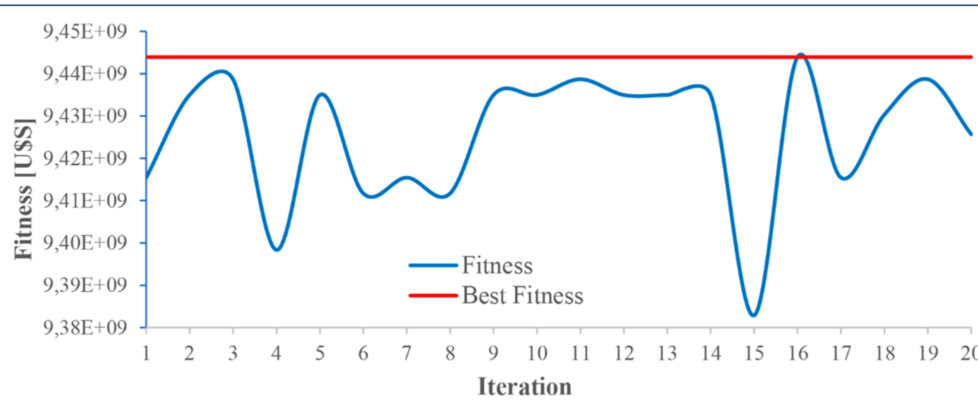


Figure 6. Fitness evolution as the iteration number increases for the standard SA procedure.

DOI: 10.1021/acs.iecr.8b02818
Ind. Eng. Chem. Res. 2018, 57, 14307–14314

Moreover, a parallel algorithm was also implemented so as to make the optimization software even more competitive. For case II, Table 3 shows the influence of parallel programming on the efficiency and accuracy of the enhanced cooperative model. The speedup measure[28] shows that for up to eight processors, parallel programming makes it possible to discover solutions of approximately equivalent quality outperforming the sequential counterpart.

Table 3. Analyzing the Effectiveness of Parallel Programming

	SA−GA−ACO (seq)	SA−GA−ACO (POWH)	SA−GA−ACO (POWH)
no. of processors	1	4	8
average fitness	9.442×10^9	9.442×10^9	9.442×10^9
average time (ms)	9322	4848	3404
speed up		1.92	2.74
efficiency		48.07	34.23

Table 4 shows the adopted settings for the parameters of the metaheuristics whose results are reported in Tables 1−3. As to GA, the mutation rate was chosen by testing its effect on the population. Mutations should be introduced carefully because they change the individuals, perhaps ruining the population. Table 5 shows that the increase in the mutation rate decreases the number of successful hits, thus augmenting the computational time required to find the best solution.

On completing their task, the optimizers returned the 30 best solutions found and the corresponding best routes, which are depicted in Figures 7 and 8. For a small-size problem (Figure 7), SA yielded the best solution with satisfactory computational times (527 ms). In contrast, Figure 8 shows the best solution for a large-size problem that was efficiently found in 2432 ms by POWH.

Table 5. Effect of Mutation Rates on GA's Performance

	GA		
mutation rate	fitness (US$)	time (ms)	successful hits (%)
0.1	9.444×10^9	19345	93
0.5	9.443×10^9	18368	87
0.99	9.443×10^9	20623	83

Figure 7. Case I: small instance. The best solution yielded by SA.

7. CONCLUSIONS

The rigorous design of pipeline networks is a timely topic, and its strong requirements of computational time have undoubtedly been detected. Therefore, it is challenging to devise novel algorithms that speed up calculations. Agile implementations with the help of modern technologies, like parallel processing, may help to develop production ideas and take them into definite shape for the industry.

In view of this issue, a parallel hyperheuristic algorithm for the design of pipeline networks is proposed in this work. This optimization method allows the designers not only to speed up the computations but also to improve the quality of the provided solutions. The search was diversified by enlarging the search space by means of GA and ACO, the latter being an efficient explorer. In this parallel cooperative A-team model, different metaheuristics exchange information related to the search with the purpose of computing better solutions.

In this paper, a realistic industrial problem of the optimal design of subsea pipeline networks was selected to carry out detailed computational experiments. The incorporation of metaheuristics into the cooperative team was analyzed step by step. During the tests, the proposed hyperheuristics exhibited outstanding performance because sharing good solutions made the search process faster.

The three metaheuristics selected as agents in this work have different complementary characteristics as regards their efficiency and robustness. Testing showed that the A-team is a proper way of organizing the algorithms to obtain solutions that could not be found by any algorithm working alone. For example, SA could find suboptimal solutions for a large instance. If a different combination had been chosen at that stage, like cooling more gradually, the solution would have been improved very slowly. In short, the main disadvantage of

Table 4. Parameter Settings and Termination Criteria (TC)

metaheuristics					
SA		GA		ACO	
evaluations (TC)	1000	generations (TC)	250	ants	250
Markov-chain length	10	individuals	50	iterations	25
temperature decay	0.99	crossover rate	0.7	colonies (TC)	150
		mutation rate	0.1	pheromone evaporation	0.8
				pheromone exponent	2
				heuristic exponent	1

DOI: 10.1021/acs.iecr.8b02818
Ind. Eng. Chem. Res. 2018, 57, 14307−14314

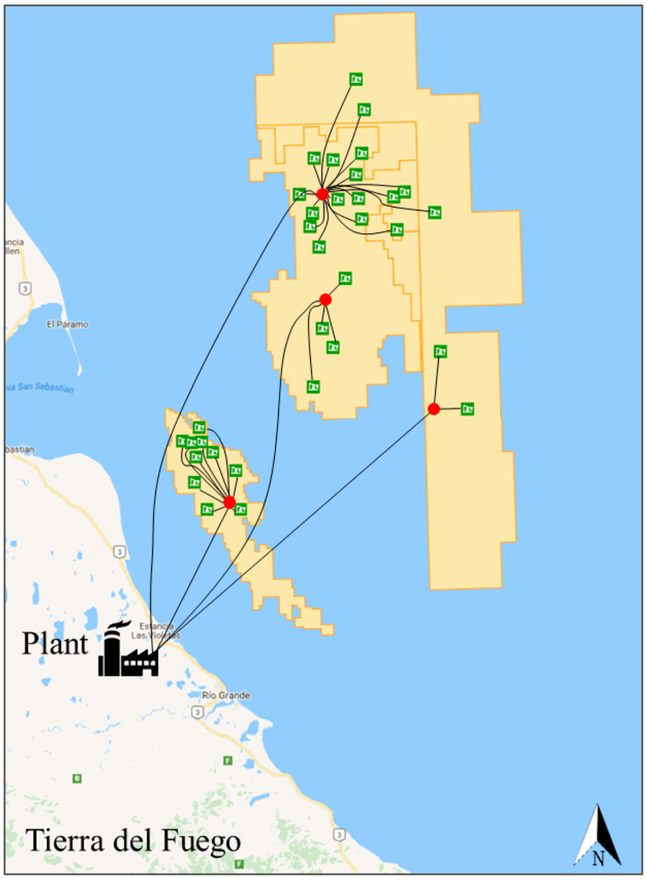


Figure 8. Case II: large instance. The best solution yielded by POWH.

this policy is that it may lead to prohibitive time delays. In contrast, GA helped effectively in the exploration, contributing to diversify the search. In this way, better solutions could be found more quickly.

As part of our future work, it would be interesting to implement a complementary parallel search for the settings of each individual metaheuristic in order to make the parameter choice automatic. In this way, the optimizer might be generalized straightaway for broader fields of application.

AUTHOR INFORMATION

Corresponding Author

*E-mail: dybrigno@criba.edu.ar. Tel: 054-0291-4861700, Int 250.

ORCID

Paola P. Oteiza: 0000-0002-1138-5626

Nélida B. Brignole: 0000-0002-4795-2872

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the financial support from the Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CIC) (Argentina) for the Proyectos De Innovación y Transferencia En Areas Prioritarias 2016 grant (PIT-AP-BA).

NOMENCLATURE

Bf_k = best solution yielded by the kth agent
Bt_k = shortest time demanded by an execution of the kth agent
$\mathrm{CC}_{ij,t}$ = construction cost from the ith to the jth point, in the tth period
Fe_k = fitness evaluation of the kth agent
FP_{ij} = feasible path from the ith to the jth point
IND_k = ranking index of the kth agent
l = total number of agents working cooperatively
LC_t = operating labor cost in the tth period
m = total number of time periods related to WP construction
$\mathrm{MC}_{k,t}$ = maintenance cost of the kth WP structure during the tth period
n = total number of pipeline construction time periods
o = project lifetime
Oe_k = optimality evaluation of the kth agent
p = number of plants onshore
P_t = sale price of gas in the tth period
q = number of potential well platforms
Q_t = amount of fluid transported in the tth period
r = discount rate
R_k = amount of times that the kth agent was repeated without registering any improvements in the solution
Re_k = repetition evaluation of the kth agent
Rq_k = number of executions performed by the kth agent
Sf_k = sum of all the best fitness values obtained by the kth agent
St_k = runtimes required by the past executions of the kth agent
t = planning horizon
Te_k = time evaluation of the kth agent
WP_k = state of the kth well platform: $\mathrm{WP}_k = 1$ if active, otherwise $\mathrm{WP}_k = 0$
δ_k = installation cost of the kth well platform
z = number of wells

REFERENCES

(1) van den Heever, S., A.; Grossmann, I.; Vasantharajan, S.; Edwards, K. A. Lagrangean Decomposition Heuristic for the Design and Planning of Offshore Hydrocarbon Field Infrastructures with Complex Economic Objectives. *Ind. Eng. Chem. Res.* **2001**, *40*, 2857−2875.

(2) Tarhan, B.; Grossmann, I. E.; Goel, V. Stochastic programming approach for the planning of offshore oil or gas field infrastructure under decision- dependent uncertainty. *Ind. Eng. Chem. Res.* **2009**, *48* (6), 3078−3097.

(3) Ulstein, N. L.; Nygreen, B.; Sagli, J. R. Tactical planning of offshore petroleum production. *Eur. J. Oper. Res.* **2007**, *176*, 550−564.

(4) Borraz-Sánchez, C. *Optimization methods for pipeline transportation of natural gas*; University of Bergen: Norway, Norway, 2010.

(5) Ríos-Mercado, R. Z.; Borraz-Sánchez, C. Optimization problems in natural gas transportation systems: A state-of-the-art review. *Appl. Energy* **2015**, *147*, 536−555.

(6) Baioco, J. S.; Coutinho, D. C. S.; Albrecht, C. H.; de Lima, B. S. L. P.; Jacob, B. P.; Rocha, D. M.; Cardoso, C. O. Synthesis and optimization of submarine pipeline routes. *Proceedings of the 10th World Congress on Computational Mechanics*; IACM: Sao Paulo, Brazil, 2012.

(7) de Lucena, R. R.; Baioco, J. S.; de Lima, B. S. L. P.; Albrecht, C. H.; Jacob, B. P. Optimal design of submarine pipeline routes by genetic algorithm with different constraint handling techniques. *Adv. Eng. Softw.* **2014**, *76*, 110−124.

DOI: 10.1021/acs.iecr.8b02818
Ind. Eng. Chem. Res. 2018, 57, 14307−14314

(8) Rocha, D. M.; Cardoso, C. O.; Borges, R. G.; Baioco, J. S.; da Costa, D.; Coutinho, S.; Albrecht, C. H.; Jacob, B. P. Optimization of submarine pipeline routes considering slope stability. *Offshore Technology Conference*; OTC: 2015.
(9) Baioco, J. S.; Albrecht, C. H.; de Lima, B. S. L. P.; Jacob, B. P.; Rocha, D. M. Multi-Objective Optimization of Subsea Pipeline Routes in Shallow Waters. *ASME 2017 36th International Conference on Ocean, Offshore and Arctic Engineering*; American Society of Mechanical Engineers: 2017; pp 1−13.
(10) Burke, E. K.; Gendreau, M.; Hyde, M.; Kendall, G.; Ochoa, G.; Özcan, E.; Qu, R. Hyper-heuristics: A survey of the state of the art. *J. Oper. Res. Soc.* **2013**, *64* (12), 1695−1724.
(11) Sabar, N. R.; Ayob, M.; Kendall, G.; Qu, R. The automatic design of a hyper-heuristic framework with gene expression programming for combinatorial optimization problems. *IEEE Trans. Evolutionary Computation.* **2015**, *19* (3), 309−325.
(12) Marshall, R. J.; Johnston, M.; Zhang, M. A comparison between two evolutionary hyper-heuristics for combinatorial optimization. In *Asia-Pacific Conference on Simulated Evolution and Learning*; Springer: Cham, 2014; pp 618−630 .
(13) Chen, Y.; Mourdjis, P.; Polack, F.; Cowling, P.; Remde, S. Evaluating hyperheuristics and local search operators for periodic routing problems. In *European Conference on Evolutionary Computation in Combinatorial Optimization*; Springer: Cham, 2016; pp 104−120.
(14) Branke, J.; Nguyen, S.; Pickardt, C. W.; Zhang, M. Automated design of production scheduling heuristics: A review. *IEEE Trans. Evol. Comput.* **2016**, *20* (1), 110−124.
(15) Pacheco, P. S. *An Introduction to Parallel Programming*; Elsevier: 2011.
(16) Gupta, V.; Grossmann, I. E. An Efficient Multiperiod MINLP Model for Optimal Planning of Offshore Oil and Gas Field Infrastructure. *Ind. Eng. Chem. Res.* **2012**, *51*, 6823−6840.
(17) Talukdar, S.; Baerentzen, L.; Gove, A.; De Souza, P. Asynchronous teams: Cooperation schemes for autonomous agents. *J. Heuristics.* **1998**, *4* (4), 295−321.
(18) de Souza, P. S.; Talukdar, S. N. Asynchronous organizations for multi-algorithm problems. In *Proceedings of the 1993 ACM/SIGAPP symposium on Applied computing: states of the art and practice*; ACM: 1993; pp 286−293.
(19) Gupta, V.; Grossmann, I. E. Development Planning of Offshore Oilfield Infrastructure. In *Alternative Energy Sources and Technologies*; Springer: Cham, 2016; pp 33−87.
(20) Coello Coello, C. A.; Lamont, G. B.; Van Veldhuizen, D. A. *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd ed.; Springer: New York, 2007.
(21) Echarte, R. *Equipos para Procesos Químicos*; EdiUNS: Argentina, 2010.
(22) Remde, S.; Cowling, P.; Dahal, K.; Colledge, N.; Selensky, E. An empirical study of hyper-heuristics for managing very large sets of low level heuristics. *J. Oper. Res. Soc.* **2012**, *63* (3), 392−405.
(23) Ministerio de Energía y Minería. Información geográfica. http://sig.se.gob.ar/visor/visorsig.php?t=4 (accessed February 13, 2018).
(24) Oteiza, P. P.; Rodriguez, D. A.; Brignole, N. B. Parallel cooperative optimization through hyperheuristics. In*Computer-Aided Chemical Engineering*, 1st ed.; 13th International Symposium on Process Systems Engineering - PSE; Elsevier: Amsterdam, 2018; Vol. *44*, pp 805−810.
(25) Rodriguez, D. A.; Oteiza, P. P.; Brignole, N. B. Hiperheurística para el planeamiento del transporte público. *Entre Ciencia e Engeniería.* **2018**, *12* (23), 103−108.
(26) Ulrich, G. D.; Vasudevan, P. T. *Chemical engineering process design and economics. a practical guide*, 2nd ed.; Process Publishing: Durham, NH, 2004, Chapter 5.
(27) ENAP. Nuestras Operaciones. http://www.enap.com.ar/nuestras-operaciones-pioneros-offshore.php (accessed February 24, 2018).
(28) Barr, R. S.; Hickman, B. L. Reporting Computational Experiments with Parallel Algorithms: Issues, Measures, and Experts Opinions. *ORSA J. Comput.* **1993**, *5* (1), 2−18.

DOI: 10.1021/acs.iecr.8b02818
Ind. Eng. Chem. Res. 2018, 57, 14307−14314