

A HYBRID OPTIMIZATION ALGORITHM FOR SOLVING CONSTRAINED ENGINEERING DESIGN PROBLEMS

Elsayed M. Zaki

Department of Basic Engineering Science, Faculty of Engineering , Shebin El-Kom, Minoufia University, Egypt .

ABSTRACT

Abstract - An algorithm called ACO-FA is proposed which incorporates concepts from ant colony optimization (ACO) and firefly algorithm (FA). The methodology of the proposed algorithm is introduced based on a refining mechanism of ACO and FA for updating the solutions of ACO-FA. Moreover, the evolutions of these solutions are performed by using feasibility rule that maintain the solutions with lower degrees of constraint violations. The proposed algorithm ACO-FA is tested on several benchmark functions. Numerical comparisons with different hybrid meta-heuristics demonstrate its effectiveness and efficiency.

Index Terms - Ant colony optimization; Firefly algorithm; constrained optimization

I. INTRODUCTION

Traditional optimization methods can be classified into two distinct groups; direct and gradient-based methods. In direct search methods, only objective function and constraint value are used to guide the search, whereas gradient-based methods use the first and/or second-order derivatives of the objective function and/or constraints to guide the search process. Since derivative information is not used, the direct search methods are usually slow, requiring many function evaluations for convergence. For the same reason, they can be applied to many problems without a major change of the algorithm. On the other hand, gradient based methods quickly converge to an optimal solution, but are not efficient in non-differentiable or discontinuous problems. In addition, if there is more than one local optimum in the problem, the result may depend on the selection of an initial point, and the obtained optimal solution may not necessarily be the global optimum. Furthermore, the gradient search may become difficult and unstable when the objective function and constraints have multiple or sharp peaks. The computational drawbacks of existing numerical methods have forced researchers to rely on meta-heuristic algorithms based on simulations to solve engineering optimization problems. The class of meta-heuristic algorithms includes but is not restricted to- particle swarm optimization (PSO), evolutionary computation (EC) including genetic algorithms (GAS), simulated annealing (SA), tabu search (TS) and particle swarm (PS). For books and surveys on meta-heuristics see [1, 2,3].

Recently, a skilled combination of meta-heuristic with other optimization techniques, a so called hybrid meta-heuristic, can provide a more efficient behaviour and a higher flexibility when dealing with real-world and large scale problems. For the above reasons, hybrid meta-heuristics currently enjoy an increasing interest in the optimization community. However, the field of hybrid meta-heuristics is still in its early days. A substantial amount of further research is necessary in order to develop clearly structured hybrid meta-heuristics that can be generally used for optimization.

In this paper, we are interested in hybrid two different forms of hybrid algorithms (ACO-FA), specifically based on the ACO and FA for solving constrained engineering design problems. The proposed algorithm integrates the merits of both ACO and FA as follows: First, the proposed algorithm is parallelized by ants and fireflies in the first stage to enhancement the search by using the characteristics of the search of both algorithms. Second, the diversity of the solutions, where, if any of the swarms are going to be in equilibrium, the evolution process will be stagnated as time goes on. Thus, ACO and/or FA do not possess the ability to improve upon the quality of the solutions as the number of generations is increased. Finally, the proposed algorithm ACO-FA is tested on several benchmark functions. The comparisons demonstrated the effectiveness and efficiency of the proposed algorithm in terms of the quality of the resulting solutions.

The organization of the remaining paper is as follows. In Section II we describe some preliminaries on optimization problems. In Sections III, ACO and FA are briefly introduced. In Section IV, hybridizing ant colony optimization with firefly algorithm, named ACO-FA, is proposed and explained in detail. Experiments and discussions are presented in Section V. Finally, we conclude the paper in Section VI.

II. PRELIMINARIES

Constrained optimization problems (COPs) are always inevitable in many science and engineering disciplines. The COPs can be formulated as in (1):

$$\min_{\Omega} F(\mathbf{x}), \mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathfrak{R}^n.$$

$$\Omega = \left\{ \mathbf{x} \mid \begin{array}{l} g_k(\mathbf{x}) \leq 0; h_j(\mathbf{x}) = 0; \\ k = 1, 2, \dots, L; j = L+1, \dots, P \end{array} \right\} \quad (1)$$

where $F(\mathbf{x})$ is the objective function defined on the search space S , $S \subset \mathbf{R}^n$, \mathbf{x} is called a decision variable and $g_k(\mathbf{x})$, $h_j(\mathbf{x})$ are defined the inequality and equality constraints, respectively. The set $\Omega \subseteq S$ and $\phi = S - \Omega$ define the feasible and infeasible search spaces, respectively. Usually, the search space S is defined as an n -dimensional rectangle in \mathbf{R}^n as in (2) (domains of variables defined by their lower and upper bounds):

$$x_j^L \leq x_j \leq x_j^U, j = 1, 2, \dots, n \quad (2)$$

Any point $\mathbf{x} \subseteq \Omega$ is called a feasible solution, otherwise, \mathbf{x} is an infeasible solution.

III. THE BASICS OF ACO AND FA ALGORITHMS

A. The Basics of ACO

Ant Colony Optimization (ACO) algorithms were introduced by Marco Dorigo [4] in the early 1990s. The ACO is a meta-heuristic approach to solve problems that has been inspired by the ants' social behaviors in finding shortest paths. Real ants walk randomly until they find food and return to their nest while depositing pheromone on the ground in order to mark their preferred path to attract other ants to follow [5]. If other ants travel along the path and find food, they will deposit more pheromone as to reinforce the path for more ants to follow [5]. Currently, the ACO has also been applied to continuous problems, where interesting results are discovered [6,7,8,9].

The whole ACO algorithm can be described by taking the travelling salesman problem (TSP) as an example. The TSP is to find a minimal length with each city visited once. We are given a set of N cities, represented by nodes, and a set E of edges with fully connecting nodes N . Let d_{ij} be the length of the edge $e_{ij} \in E$, which is the distance between cities i and j . At each iteration t , an ant in city i has to choose the next city j to head for from among those cities that it has not yet visited. The probability of picking a certain city j is calculated using the distance between cities i and j , and the amount of pheromone on the edge between these two cities. Based on this TSP, initially, all the agents (ants) are randomly dispersed across the cities space. The two phases of the ACO algorithm are as follows.

1) *Solution construction rule*: the probability with which an ant a chooses to go from city i to city j is described as in (3):

$$p_{ij}^a(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^a} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta} & \text{if } \exists j \in N_i^a \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where $\tau_{ij}(t)$ is the amount of pheromone trails on edge $e_{ij} \in E$ at iteration t , $\eta_{ij} = 1/d_{ij}$ is the heuristic value of moving from city i to city j , N_i^a is the set of neighbours of city i for the a^{th} ant, and parameters α and β controls the relative weight of pheromone trail and heuristic value, respectively.

Thus an ant a at city i chooses the city j to go to with a probability p_{ij}^a , which is a function of the town distance and of the amount of pheromone trail present on the connecting edge.

2) *Depositing the pheromone rule*: each ant constructs a tour by repeatedly applying the solution construction rule, and the ant also updates the amount of pheromone on the visited edges by applying the global updating rule [4] using (4):

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \sum_{a=1}^m \nabla \tau_{ij}^a(t), \quad t = 1, 2, \dots, T \quad (4)$$

$$\nabla \tau_{ij}^a = \begin{cases} C/L^a & \text{if } \exists \text{ ant } a \text{ travel on } e_{ij} \\ 0 & \text{otherwise} \end{cases}$$

where L^a is the length of a^{th} ant journey, and C is a constant parameter, $\rho \in (0,1)$ is the rate of the pheromone evaporation, m is the number of ants, T is the total is the number of iterations and $\nabla \tau_{ij}^a$ is the amount of pheromone deposited by a^{th} ant. The pseudo code of the ACO can be summarized in the Fig. 1

B. The Basics of FA

FA [10] is one of the most recent meta-heuristic techniques for approximate optimization. FA is inspired by the firefly's biochemical and social aspects and based upon the following three assumptions: (a) Each firefly attracts all the other fireflies with weaker flashes; (b) Attractiveness is proportional to their brightness which is reverse proportional to their distances; (c) No fireflies can attract the brightest firefly, and it moves randomly.

FA adopts some social terminologies, including: the variation of light intensity and formulation of attractiveness. Yang [10] simplified that the attractiveness of a firefly is determined by its brightness which in turn is associated with the objective function. In general, the attractiveness is proportional to their brightness. Furthermore, every member of the firefly swarm is characterized by its brightness that can be directly expressed as an inverse of an objective function for a minimization problem. Based on this objective function, initially, all the agents (fireflies) are randomly dispersed across the search space. The two stages of the firefly algorithm are as follows.

1) *Variation of light intensity*: Light intensity is related to objective values [10]. So for a maximization/minimization problem a firefly with high/low intensity will attract another firefly with high/low intensity. Assume that there exists a swarm of m agents (fireflies) and \mathbf{x}_i represents a solution for a firefly i , whereas $f(\mathbf{x}_i)$ denotes its fitness value. Here the brightness I of a firefly is selected to reflect its current position \mathbf{x} of its fitness value $f(\mathbf{x})$ [10], given as in (5):

$$I(\mathbf{x}_i) = f(\mathbf{x}_i), i = 1, 2, \dots, m \quad (5)$$

2) *Movement toward attractive firefly*: The firefly has an attractiveness which is proportional to the light intensity seen by adjacent fireflies. Each firefly has its distinctive attractiveness βf which implies how strong it attracts other members of the swarm. However, the attractiveness βf is relative; it will vary with the distance r_{ij} between two fireflies i and j at locations \mathbf{x}_i and \mathbf{x}_j respectively, is given as in (6):

$$r_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\| \quad (6)$$

The attractiveness function $\beta f(r)$ of the firefly is determined by using (7):

$$\beta f(r) = \beta f_0 e^{-\gamma r^2} \quad (7)$$

where βf_0 is the attractiveness at $r = 0$ and γ is the light absorption coefficient.

The movement of a firefly i at location \mathbf{x}_i attracted to another more attractive (brighter) firefly j at location \mathbf{x}_j is determined as in (8):

$$\mathbf{x}_i(t+1) = \mathbf{x}_i + \beta f_0 e^{-\gamma r^2} (\mathbf{x}_j - \mathbf{x}_i) + \alpha_f (rand - 0.5) \quad (8)$$

where the second term is due to the attraction while the third term is randomization with α_f being the randomization parameter and $rand$ is a random number generator uniformly distributed in $[0,1]$. The pseudo code of the FA can be summarized in the Fig. 2.

```

Set parameters.
Initialize the pheromone on each edge.
while termination conditions not met do
  for each ant do
    Select the start city randomly.
    Construct the solution by using (3).
  end for
  Determine the length of the tour
  Update and deposit pheromone on trails by using (4).
end while

```

Fig. 1 The pseudo code of the ACO

```

Initialize a population of fireflies within the search space.
while the condition not terminated do
    Evaluate the fitness of the population  $F(\mathbf{x})$  .
    Calculate light intensity  $I(\mathbf{x})$  as in (5).
    Rank the light intensity.
    for each firefly do
        Move the less bright firefly towards the brighter      by using (11).
        Vary attractiveness with distance  $r$  via  $e^{-r^2}$ 
    end do
end while

```

Fig. 2 The pseudo code of the FA

IV. THE PROPOSED ALGORITHM

In this section, ACO-FA is introduced in detail. In order to handle the constraints, we minimize the original objective function as well as the degree of constraint violation. Two kinds of populations with the same size m are used. In the initial step of the algorithm, populations (denoted by pop1 for ACO and pop2 for FA) are created randomly. At each generation, ACO and FA generate solutions in parallel and the resulting solutions by using two populations are sorted according to the degree of constraint violation in a descending order. Therefore, only the first half of each population are compared based on feasibility rule that determined the reflected back solution into the two populations but the second half of two populations are also interrelated. The main steps of the ACO-FA are summarized as follows:

Step 1: for this step, pop1 is created randomly in the search space. For every solution of the ant, the pheromone weighting $\tau_i(0)$ is attached to it.

Step 2: this step is devoted to evaluate the solution of pop1. The most popular technique to handle constraints is to use penalty functions that penalize infeasible solutions by reducing their fitness values in proportion to their degrees of constraint violation [11]. Only inequality constraints will be considered in this work, because equality constraints can be transformed into inequalities using: $|h_k(\mathbf{x}) - \delta| \leq 0$ and δ is a small positive number. We selected $\delta = 10^{-8}$. In particular, the i^{th} ant in ACO is evaluated by using (9):

$$F_t(\mathbf{x}_i) = F(\mathbf{x}_i) + \omega(\mathbf{x}_i), \quad (9)$$

$$\omega(\mathbf{x}_i) = \frac{1}{\mu} \left(\sum_{l=1}^L [\max(0, g_l(\mathbf{x}_i))]^\psi + \sum_{j=1}^P \sum_{l=1}^{L_j} [\max(0, |h_j(\mathbf{x}_i)| - \delta)]^\varphi \right)$$

where $F(\mathbf{x}_i)$ is the objective value of the i^{th} ant, ψ, φ is 1 or 2 and $\mu > 0$. By driving μ to zero, we penalize the constraint violations with increasing severity. After evaluating each ant according to $F_t(\mathbf{x}_i)$, the pheromone is updated by using (4) by replacing τ_{ij}^a by τ_i^a (pheromone associated with solution) and also replacing L^a by $F_t(\mathbf{x}_i)$. On the other hand η_{ij} is replaced by $\eta_i = 1/F(\mathbf{x}_i)$

Step 3: Once the pheromone is updated after an iteration, the next iteration starts by changing the ants' solutions in a manner that respects pheromone concentration. For each ant and for each dimension construct a new candidate groups to replace the old one by using (3) and replacing τ_{ij} by τ_i .

Step 4: Initialize pop2 of fireflies in the search space. And calculate the light intensity for each firefly as in (5).

Step 5: The movement of a firefly is formulated as mentioned earlier in (8).

Step 6: After ACO and FA generate solutions in parallel, the resulting solutions by two the populations sorted according to the degree of constraint violation in a descending order then the feasibility rule is applied to the first half of the two populations by comparing pair-wise solutions. The pair-wise solutions are presented by associating one from ACO onto one from FA. The resulting feasible solution is taken for updating the two populations and the infeasible solution is considered by taking the minimum violation of constraints. The idea of feasibility rule is described in Fig. 3.

After applying the feasibility rule the solutions obtained replace the first half of the two populations and the steps from 2 to 7 are repeated until the maximum iteration is reached or the best solution is obtained.

V. EXPERIMENTAL RESULTS

In this section, we will carry out numerical simulation based on some well-known constrained optimization problems to investigate the performances of the proposed ACO-FA. The selected problems have been well studied before as benchmarks by various approaches. We will also compare our results with some good results previously reported by the literature to demonstrate the efficiency and robustness of the proposed ACO-FA approach. The algorithm is coded in MATLAB 7 and is run on a Pentium IV 2.4 GHz processor with 1.0 GB RAM.

A. Parameters setting

The proposed algorithm contains number of parameters. These parameters affect the performance of the proposed algorithm. Extensive experimental tests were conducted to see the effect of different values on the performance of the proposed algorithm. Based upon these observations, the following parameters have been set as in Table I.

Input: $pop1 = pop1/2$; $pop2 = pop2/2$.
Set $i = 0$
 $\mathbf{x}_i \in pop1 \wedge \forall \mathbf{x}' \in pop2$
If $\mathbf{x}_i \wedge \mathbf{x}' \in \Omega$, then $\mathbf{x} = \min F(\mathbf{x}' OR \mathbf{x}_i)$
Else if $\mathbf{x}_i \wedge \mathbf{x}' \notin \Omega$, then $\mathbf{x} = \min \omega(\mathbf{x}' OR \mathbf{x}_i)$
Else if $\mathbf{x}_i \in \Omega \wedge \mathbf{x}' \notin \Omega$ then $\mathbf{x} = \mathbf{x}_i$
End if
 $i = i + 1$
output : \mathbf{x}

Fig.3 The pseudo code of feasibility rule

B. Benchmark functions

There are many different test functions for COPs [3,12]. To validate the proposed ACO-FA, we have selected a subset of these functions. To be more specific in this paper, we have tested the following five functions:

- The first test function can be stated as:

$$\min F_1(\mathbf{x}) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 \\ + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$$

subject to :

$$g_1(\mathbf{x}) = 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 - 127 \leq 0 \quad (11)$$

$$g_2(\mathbf{x}) = 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 - 282 \leq 0$$

$$g_3(\mathbf{x}) = 23x_1 + x_2^2 + 6x_6^2 - 8x_7 - 196 \leq 0$$

$$g_4(\mathbf{x}) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leq 0$$

$$-10 \leq x_i \leq 10, i = 1, 2, \dots, 7.$$

- The second test function can be stated as:

$$\min F_2(\mathbf{x}) = x_1 + x_2 + x_3$$

subject to :

$$g_1(\mathbf{x}) = -1 + 0.0025(x_4 + x_6) \leq 0$$

$$g_2(\mathbf{x}) = -1 + 0.0025(x_5 + x_7 - x_4) \leq 0$$

$$g_3(\mathbf{x}) = -1 + 0.01(x_8 - x_5) \leq 0 \quad (13)$$

$$g_4(\mathbf{x}) = -x_1x_6 + 833.33252x_4 + 100x_1 - 83333.333 \leq 0$$

$$g_5(\mathbf{x}) = -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \leq 0$$

$$g_6(\mathbf{x}) = -x_3x_8 + x_3x_5 - 2500x_5 + 1250000 \leq 0$$

$$100 \leq x_1 \leq 10000, 1000 \leq x_2, x_3 \leq 10000$$

$$10 \leq x_i \leq 1000 \quad i = 4, 5, \dots, 8.$$

TABLE I
PARAMETERS SETTING

The size of ant colony (m)	100
Initial pheromone (τ_0)	20
Evaporation rate (ρ)	0.9
The constant of ACO (C)	400
Pheromone weight (α)	1
Heuristic value (β)	1
Initial light intensity (I_0)	0
Initial attractiveness (β_{f0})	1
The light absorption coefficient (γ)	1
The randomness reduction constant (θ)	0.9
Randomization parameter (α_f)	0.2

- The third test function can be stated as:

$$\begin{aligned} \min F_3(\mathbf{x}) = & x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 \\ & + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 \\ & + (x_{10} - 7)^2 + 45 \end{aligned}$$

subject to :

$$\begin{aligned} g_1(\mathbf{x}) &= -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \leq 0 \\ g_2(\mathbf{x}) &= 10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0 \\ g_3(\mathbf{x}) &= -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \leq 0 \quad (14) \\ g_4(\mathbf{x}) &= 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \leq 0 \\ g_5(\mathbf{x}) &= 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \leq 0 \\ g_6(\mathbf{x}) &= x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \leq 0 \\ g_7(\mathbf{x}) &= 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30 \leq 0 \\ g_8(\mathbf{x}) &= -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leq 0 \\ & -10 \leq x_i \leq 10, i = 1, 2, \dots, 10. \end{aligned}$$

- The fourth test function can be stated as:

$$\min F_4(\mathbf{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$$

subject to :

$$\begin{aligned} g_1(\mathbf{x}) &= 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.002205x_3x_5 \\ g_2(\mathbf{x}) &= 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 \\ g_3(\mathbf{x}) &= 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.00190853x_3x_4 \\ 0 &\leq g_1(\mathbf{x}) \leq 92, 90 \leq g_2(\mathbf{x}) \leq 110, 20 \leq g_3(\mathbf{x}) \leq 25, \\ 78 &\leq x_1 \leq 102, 33 \leq x_2 \leq 45, 27 \leq x_3 \leq 45, i = 3, 4, 5. \end{aligned}$$

The obtained results by the proposed ACO-FA for test functions are presented in Table II compared with the previous best solutions reported by the other algorithms. The obtained results demonstrate the effectiveness and efficiency of the proposed ACO-FA for solving constrained optimization problems and superiority of the solutions for all test functions.

B) Optimization problems

Design optimization, especially design of electrical devices, has many applications in engineering and industry. Thus, we have chosen the linear synchronous motor (LSM) and air-cored solenoid as constrained engineering design problems [13,14].

B.1) Design of LSM

The linear synchronous motor (LSM) operates on the same working principle as that of a permanent magnet rotary D.C. motor [13]. As in a rotary motor there are two parts in a LSM, one is the set of permanent magnets and the other is the armature

that has conductors carrying current. The permanent magnets produce a magnetic flux perpendicular to the direction of motion. The flow of current is in the direction perpendicular to both the direction of the motion and the direction of the magnetic flux.

To validate the proposed algorithm, it is employed to optimize the geometrical design of the linear electric actuator problem as is shown in Fig. 4. The objective function is to maximum force subject to some of constraints on heat, radius, saturation, demagnetization and maximum force constraint. There are four design variables: the current in each slot, x_1 , the dimensions of the slot, x_2 and x_3 , the height of the magnet, x_4 . The mathematical formulation of the objective function is described as follows using (15):

TABLE II
THE OBTAINED RESULTS

Test function	F_1	F_2	F_3	F_4
Optimal solution	680.6300573	7049.3307	24.3062091	-
ACO-FA	680.63755	7053.77	24.3367	-30665.533
Michalewicz [3]	680.642	7377.976	24.690	-
Lee and Geem [12]	680.6413574	7057.274414	24.3667946	-30665.500

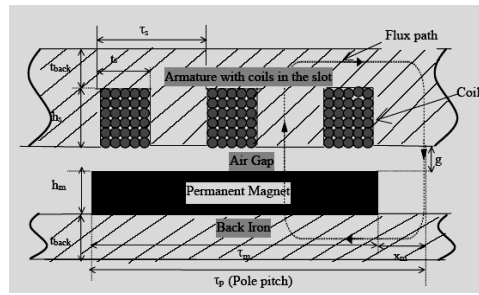


Fig. 4 Cross-section of approximately a one pole pitch long section of linear synchronous motor

$$\max F_5(\mathbf{x}) = 653.45x_1x_4(0.02825 + x_4 + x_2/2)/(x_4 + 0.00117)$$

subject to:

$$g_1(\mathbf{x}) = 1.92 * 10^{-5} x_1^2 (0.02825 + x_4 + x_2/2) / x_2 x_3 - 4000 \leq 0$$

$$g_2(\mathbf{x}) = x_4 + x_2 - 3x_3 - 0.0266 \leq 0 \quad (15)$$

$$g_3(\mathbf{x}) = 1.3x_4 / (0.00117 + x_4) + 165.13x_3 - 1.5 \leq 0$$

$$g_4(\mathbf{x}) = 6.28 * 10^{-7} x_1 (0.00105 + x_4) / x_4 (x_4 + 0.001) - 1.17 \leq 0$$

$$g_5(\mathbf{x}) = 15000 - 653.45x_1x_4(0.02825 + x_4 + x_2/2)/(x_4 + 0.00117) \leq 0$$

$$0 \leq x_1 \leq 1028A, 0 \leq x_2, x_4 \leq 0.08m, 0 \leq x_3 \leq 0.009m.$$

The objective function value obtained by our method is $F_4 = 16236$ N and the objective function value obtained by Deshpande is $F_4 = 16192$ N [13]. Therefore the ACO-FA result is superior to those obtained using the Deshpande. Also the decision variables found by the ACO-FA is $\mathbf{x} = (368.4769A, 1.1 \text{ mm}, 32.8 \text{ mm}, 26.1 \text{ mm})$

B.2) Design of an air-cored solenoid

The optimization problem of a coreless solenoid with rectangular cross section $x_1 \times x_2$ and a mean radius x_3 (see Fig. 5) is tackled from [14]. This problem can then be formally defined in the following two terms: maximize the inductance $L(x_1, x_2, x_3)$ and satisfy some of constraints as in (16) for the given length $k_1 = 10m$ and $k_2 = 10^{-6} m^2$ of the current carrying wire. In order to simplify the analysis, two variables, x_1 and x_2 , are considered. Correspondingly, the optimization problem is simplified as:

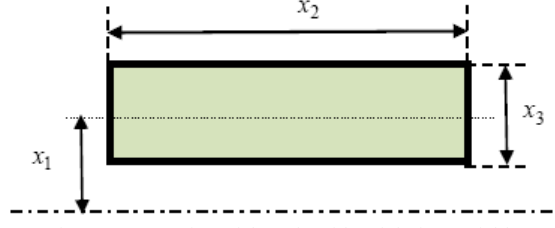


Fig.5. Cross section of the solenoid and design variables

$$\max F_6(\mathbf{x}) = 31.49 \frac{k_1^2}{4\pi^2 x_2} \left/ \left(9 + 6 \frac{x_1}{x_2} + 5 \frac{k_1 k_2}{\pi x_1 x_2^2} \right) \right.,$$

subject to:

$$g_1(\mathbf{x}) = \frac{\pi x_1^2 x_2}{4} + \frac{k_1^2 k_2^2}{4\pi x_1^2 x_2} + \frac{k_1 k_2}{2} \leq 0.0094 \quad (16)$$

$$g_2(\mathbf{x}) = x_1 - \sqrt{\frac{k_1 k_2}{4\pi x_2}} > 0, \quad x_1 \in [0, 0.1], x_2 \in [0, 0.3]$$

The obtained solution by our approach is demonstrated in Fig. 6, where the objective is increase with the increasing the number of iterations (i.e., $F_5 = 258.0526 \mu H$, $x_1 = 0.0016$, $x_2 = 0.3$).

In this subsection, a comparative study has been carried out to assess the proposed approach concerning quality of the solution. On the first hand, evolutionary techniques suffer from the quality of solution. Therefore the proposed algorithm has been used to increase the solution quality by combining the two merits of two meta-heuristic algorithms. On the other hand, unlike classical techniques our approach search from a population of points, not single point. Therefore the proposed algorithm can provide a globally optimal solution among this population. In addition, our approach uses only the objective function information, not derivatives or other auxiliary knowledge. Therefore it can deal with the non-smooth, non-continuous and non-differentiable functions which are actually existed in practical optimization problems. Another advantage is that the simulation results prove superiority of the proposed approach to those reported in the literature, where it is completely better than the other approaches. So, the ACO-FA approach is quite competitive when compared with the other existing methods. Finally, the reality of using the proposed approach to handle complex problems of realistic dimensions has been approved due to procedure simplicity.

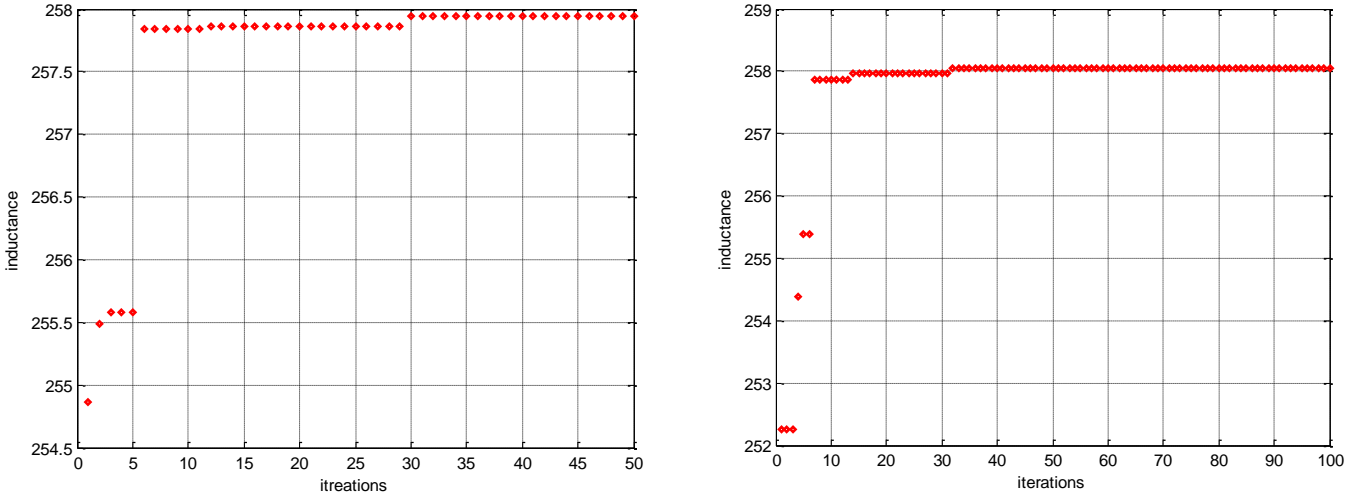


Fig.6. Maximum inductance against the iteration after 50 iterations (left) and after 100 iterations (right)

VI. CONCLUSIONS

This paper presents a hybrid approach combining two heuristic optimization techniques, ACO and FA. Our approach integrates the merits of both ACO and FA to solve COPs where, the position of ants is updated using the local search scheme. Therefore, ACO explore and exploit the spare solutions in the search space. Moreover, the fireflies are used in parallel to search good decision solutions, where the evolutions of these solutions are performed by using feasibility rule that maintain the solutions

with lower degrees of constraint violations. Therefore, the proposed algorithm speeds up the convergence and improves the algorithm's performance. The proposed algorithm ACO-FA is tested on several benchmark problems from the usual literature and two engineering optimization problems. From the comparative study, ACO-FA has shown its potential to handle various COPs, and its performance is much better than other state-of-the-art.

The future work will be focused on two directions: (i) the application of ACO-FA to real COPs from industry; and (ii) the extension of the method to solve the multi-objective problems.

REFERENCES

- [1] H. Zang, S. Zhang, K. Hapeshi. A Review of Nature-Inspired Algorithms. *Journal of Bionic Engineering* 7 Suppl. S232–S237, 2010.
- [2] F. Glover, G. Kochenberger, editors. *Handbook of metaheuristics*. Norwell, MA: Kluwer Academic Publishers; 2002.
- [3] Z. Michalewicz, Genetic algorithms, numerical optimization, and constraints, in: L. Esheman (Ed.), *Proceedings of the Sixth International Conference on Genetic Algorithms*, Morgan Kaufman, San Mateo, 151–158, 1995.
- [4] M. Dorigo, Learning and Nature Algorithm (in Italian). Ph.D. Dissertation, Dipartimento di Eletttronica, Politecnico di Milano, Italy, 1992.
- [5] M. Dorigo, E. Bonaneau, , G. Theraulaz, Ant algorithms and stigmergy. *Future Generation Computer Systems*, 16, 851–871, 2000.
- [6] A. A. Mousa, Wael F. Abd El-Wahed, R. M. Rizk-Allah, A hybrid ant colony optimization approach based local search scheme for multiobjective design optimizations, *Electric Power Systems Research*, 81 , 1014–1023, 2011.
- [7] J. Drezo, P. Siarry, An ant colony algorithm aimed at dynamic continuous optimization. *Appl. Math. Comput.*, 181, 457–467, 2006.
- [8] M. Dorigo, T. Stützle, *Ant Colony Optimization*, MIT Press, London, 2004.
- [9] M.D. Toksar, A heuristic approach to find the global optimum of function, *Journal of Computational and Applied Mathematics* 209 (2) 160 – 166, 2007.
- [10] X.S. Yang.: *Nature-Inspired Metaheuristic Algorithms*. Luniver Press (2008).
- [11] H. Qie, W. Ling, An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Engineering Applications of Artificial Intelligence* 20 ,89 – 99, 2007.
- [12] K. S. Lee, Z. W. Geem, A new meta-heuristic algorithm for continues engineering optimization: harmony search theory and practice. *Comput Methods Appl Mech Eng*;194:3902–33, 2004.
- [13] A. D. Deshpande, California Linear Drives, Inc., www.calinear.com, A Study Of Methods To Identify Constraint Dominance In Engineering Design Problems, M.S. Thesis, Mechanical and Industrial Engineering Department, University of Massachusetts, Amherst, 2002.
- [14] P. Di Barba, M. Farina, and A. Savini, “An improved technique for enhancing diversity in Pareto evolutionary optimization of electromagnetic devices,” *COMPEL*, vol. 20, pp. 482–496, 2001.