***Journal of Natural Sciences and Mathematics***
*Qassim University,* Vol. 6, No. 1, PP 1-22 (January 2013/Safar 1434H.)

# A Novel Hybrid Ant Colony Optimization and Firefly Algorithm for Solving Constrained Engineering Design Problems

**Ahmed Ahmed El-Sawy[1] , Elsayed M. Zaki [2] , R. M. Rizk-Allah[2]**

*[1] Department of Mathematics, Faculty of Science, Qassim University, KSA. E-mail: a.elsawy1954@yahoo.com*

*[2] Department of Basic Engineering Science, Faculty of Engineering, Shebin El Kom, Minoufia University, Egypt. E-mail: elsayedzaki68@yahoo.com(Elsayed M. Zaki). E-mail address: rizk_abd@yahoo.com (R. M. Rizk-Allah).*

(Received24/06/2012, Accepted01/09/2012)

**Abstract.** This paper proposes ACO-FFA for constrained optimization problems, which incorporates the concepts from ant colony optimization (ACO) and firefly algorithm (FFA). The methodology of the proposed algorithm is introduced based on a parallel mechanism of ACO and FFA for updating the solutions of ACO-FFA. 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-FFA is tested on several benchmark functions. The comparisons show that ACO-FFA outperforms or performs similarly to state-of-the-art approaches in terms of the quality of the resulting solutions.

**Keywords**. Ant Colony Optimization, Firefly Algorithm, Constrained Optimization.

## 1. 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 optimal 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 optimal. 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), and tabu search (TS). For books and surveys on meta-heuristics see [1,2,3,4].

In recent years it has become evident that the concentration on a sole meta-heuristic is rather restrictive. A skilled combination of meta-heuristic with other optimization techniques, a so called hybrid meta-heuristic, can provide a more efficient behavior 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.

Among the existing meta-heuristic algorithms, a well-known branch is the ACO which is a stochastic search procedure based on observations of social behaviors of real insects or animals. The original algorithm of ACO is known as the ant system [5] which was proposed by Dorigo to solve the traveling salesman problem. Since then, some algorithms based on the ACO are presented, such as the ant colony system, MAX-MIN Ant System and the rank-based ant system [6]. On the other hand several researchers have designed ACO algorithms to deal with the continuous problems, where interesting results are discovered [7,8,9]. Currently, the Hybridization of ACO with other global optimization techniques have been proposed [10,11,12] and hardly any paper has considered introducing the FFA algorithm [13] into ACO.

A promising new meta-heuristic algorithm denoted as FFA. FFA may be considered as a typical swarm based approach for optimization, in which the search algorithm is inspired by social behavior of fireflies and the phenomenon of bioluminescent communication. There are two important issues in the firefly algorithm that are the variation of light intensity and formulation of attractiveness. Yang [13] 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 bright that can be directly expressed as an inverse of an objective function for a minimization problem.

In this paper, we propose a novel hybrid algorithm named ACO-FFA for solving the constrained optimization problems. Our algorithm integrates the merits of both ACO and FFA and it has two characteristic features. Firstly, the 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. Secondly, feasibility rule is implemented in the second stage, where the feasibility rule maintains the solutions with lower degrees of constraint violations that are found by the ants and fireflies in the earlier stage. Finally, the proposed algorithm ACO-FFA is tested on several benchmark problems from the usual literature. Comparisons show that ACO-FFA outperforms or performs similarly to state-of-the-art approaches in terms of the quality of the resulting solutions.

The organization of the remaining paper is as follows. In Section 2 we describe some preliminaries on optimization problems. In Sections 3, ACO and FFA are briefly introduced. In Section 4, hybridizing ant colony optimization with firefly algorithm, named ACO-FFA, is proposed and explained in detail. Experiments and discussions are presented in Section 5. Finally, we conclude the paper in Section 6 and the references are presented in Section 7 .

## 2. Preliminaries

Constrained optimization problems (COPs) are always inevitable in many science and engineering disciplines. Without loss of generality, the COPs can be formulated as :

$$\min f(\underline{x}), \underline{x} = (x_1, x_2, ..., x_n) \in \mathrm{R}^n. \tag{1}$$

where $\underline{x} \in \Omega \subseteq S$. The objective function $f$ is defined on the search space $S \subseteq \mathrm{R}^n$. The set $\Omega \subseteq S$ and $\psi = S - \Omega$ define the feasible and infeasible search spaces, respectively. Usually, the search space $S$ is defined as an $n$-dimensional rectangle in $\mathrm{R}^n$ (domains of variables defined by their lower and upper bounds):

$$x_j^L \le x_j \le x_j^U, \ j = 1,2,...,n \tag{2}$$

whereas the feasible set $\Omega \subseteq S$ is defined by a set of additional $P \geq 0$ constraints:

$$\Omega = \left\{ \begin{array}{l} \underline{x} \mid g_k(\underline{x}) \leq 0;\ h_j(\underline{x}) = 0; \\ \quad k = 1,2,\ldots,L;\ j = L+1,\ldots.,P \end{array} \right\} \tag{3}$$

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

## 3. The Basic ACO and FFA Algorithms

### 3.1. Ant Colony Optimization (ACO)

Ant colony optimization (ACO) algorithms were introduced by Marco Dorigo [5] in the early 1990s. While moving, ants leave a chemical pheromone trail on the ground. Ants are guided by pheromone smell. Ants tend to choose the paths marked by the strongest pheromone concentration. The indirect communication between the ants via pheromone trails enables them to find shortest paths between their nest and food sources [14].

The whole ACO algorithm can be described by taking the traveling 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, 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. The first algorithm to fall within the ACO algorithms framework was the ant system (AS) Dorigo *et al.* [5]. In AS algorithm, the probability with which an ant $a$ chooses to go from city $i$ to city $j$ by using Equation (4):

$$p_{ij}^{a}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l \in N_i^a}[\tau_{il}(t)]^{\alpha}[\eta_{il}]^{\beta}} & \text{if } \exists\, j \in N_i^a \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

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 neighbors of city $i$ for the $a^{th}$ ant, and parameters $\alpha$ and $\beta$ controls the relative weight of pheromone trail and heuristic value, respectively. After all ants have completed their tours, the pheromone level is updated by Equation (5):

$$\tau_{ij}(t+1) = (1-\rho).\tau_{ij}(t) + \sum_{a=1}^{m} \nabla \tau_{ij}^{a}(t), \quad t = 1,2,...,T \tag{5}$$

where $\rho \in (0,1)$ 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 ant $a$, given by Equation (6):

$$\nabla \tau_{ij}^{a}(t) = \begin{cases} Q/L^{a} & \text{if } \exists \text{ ant } a \text{ travel on } e_{ij} \\ 0 & \textit{otherwise} \end{cases} \tag{6}$$

where $L^a$ is the length of the $a^{th}$ ant journey, and $Q$ is a constant parameter. The pseudo code for a basic ACO algorithm is summarized in the Figure (1).

```
Set parameters and initialize the pheromone on each edge
while termination conditions not met do
    for  each ant do
        Chose one of the nodes of the TSP graph randomly as start node
        Choose the next city with the probability given by using the Equation (4)
    end for
    For each ant determine the length of the  tour
    Update pheromone trails using Equations (5) and (6)
end while
```

**Fig. (1). The pseudo code for a basic ACO algorithm.**

### 3.2. Firefly Algorithm (FFA)

The Firefly Algorithm [13] is a population-based algorithm to find the global optima of objective functions based on swarm intelligence, investigating the foraging behavior of fireflies. In the FFA, physical entities (agents or fireflies) are randomly distributed in the search space. Agents are thought of as fireflies that carry a luminescence quality, called luciferin, that emit light proportional to this value. Each firefly is attracted by the brighter glow of other neighboring fireflies. The attractiveness decreases as their distance increases. If there is no brighter one than a particular firefly, it will move randomly.

Based on this objective function, initially, all the agents (fireflies) are randomly dispersed across the search space. The two phases of the firefly algorithm are as follows.

*I. Variation of light intensity:* Light intensity is related to objective values [13]. 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 $\underline{x}_i$ represents a solution for a firefly $i$ , whereas $f(\underline{x}_i)$ denotes its fitness value. Here the brightness $I$ of a firefly is selected to reflect its current position $\underline{x}$ of its fitness value $f(\underline{x})$ , given by Equation (7):

$$I(\underline{x}_i) = f(\underline{x}_i), i = 1,2,...,m \tag{7}$$

*II. Movement toward attractive firefly:* Firefly attractiveness is proportional to the light intensity seen by adjacent fireflies [13]. Each firefly has its distinctive attractiveness $\beta$ which implies how strong it attracts other members of the swarm. However, the attractiveness $\beta$ is relative; it will vary with the distance $r_{ij}$ between two fireflies $i$ and $j$ at locations $\underline{x}_i$ and $\underline{x}_j$ respectively, is given as in the Equation (8):

$$r_{ij} = \left\| \underline{x}_i - \underline{x}_j \right\| \tag{8}$$

The attractiveness function $\beta(r)$ of the firefly is determined by using Equation (9):

$$\beta(r) = \beta_0 e^{-\gamma r^2} \tag{9}$$

where $\beta_0$ is the attractiveness at $r = 0$ and $\gamma$ is the light absorption coefficient.

The movement of a firefly $i$ at location $\underline{x}_i$ attracted to another more attractive (brighter) firefly $j$ at location $\underline{x}_j$ is determined by Equation (10):

$$\underline{x}_i(t+1) = \underline{x}_i(t) + \beta_0 e^{-\gamma r^2} (\underline{x}_j - \underline{x}_i) + \alpha_1 (rand - 0.5) \tag{10}$$

where the second term is due to the attraction while the third term is randomization with $\alpha_1$ being the randomization parameter and $rand$ is a random number generator uniformly distributed in $[0,1]$. The pseudo code of the FFA can be summarized in the Figure (2).

Set values of parameters.
Create an initial population of fireflies, *m,* within *n*-dimensional search space $\underline{x}_i, i = 1,2,...,m$ .
Evaluate the fitness of the population $f(\underline{x}_i)$ which is directly proportional to light intensity $I(\underline{x}_i)$ .
**while** (not termination condition ) **do**
    **for** $i = 1$ *to* $m$
        **for** $j = 1$ *to* $m$ .
            **if** $I_i > I_j$ ,
                Move firefly $i$ towards $j$ by using Equation (10)
            **end if**
            Vary attractiveness with distance $r$ via $e^{-\gamma r^2}$
            Evaluate new solutions and update light intensity by using Equation (7)
        **end for** $j$
    **end for** $i$
    Rank fireflies and find the current best;
**end while**

**Fig. (2). The pseudo code of the FFA.**

## 4. The Proposed Algorithm

In this section, ACO-FFA 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 particular, one kind of a single swarm is used the ants and another kind of a single swarm is used the fireflies. In the initial step of the algorithm, populations (denoted by pop1 for ACO and pop 2 for FFA) are created randomly. At each generation, ACO and FFA 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-FFA are summarized as follows:

### Step 1: Initialization

As for a continuous problem of $n$ dimensions, the algorithm assigns a random vector $\underline{x}$ for each ant, where all the $m$ solutions for the variable $i$ represent a group of candidate values. For every variable in every ant, a pheromone weighting $\tau_0$ is attached to it. Figure (3) describes the initialization, where the thickness of the color in the circles corresponds to the amount of the pheromones associate to every variable in the candidate groups.

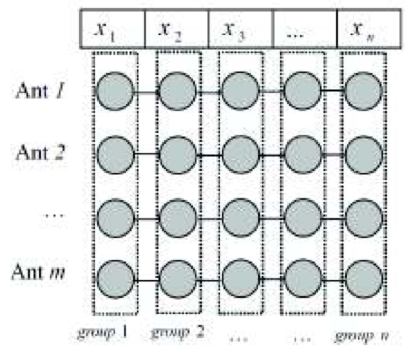


**Fig. (3). The initialization of ACO.**

**Step 2: Evaluation:**

For constrained optimization problems, the most popular approach 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 [15]. Only inequality constraints will be considered in this work, because equality constraints can be transformed into inequalities using: $|h_k(\underline{x})| - \delta \le 0$, where $\delta$ is a small positive number. We selected $\delta = 10^{-8}$. In particular, the $i^{th}$ ant in ACO is evaluated by using the Equations (11) and (12):

$$F(\underline{x}_i) = f(\underline{x}_i) + \omega(\underline{x}_i)\,, \tag{11}$$

$$\omega(\underline{x}_i) = \frac{1}{\mu}\left(\sum_{l=1}^{L}\left[\max(0, g_l(\underline{x}_i))\right]^{\psi} + \sum_{j=1}^{P-L}\left[\max(0, |h_j(\underline{x}_i)| - \delta)\right]^{\varphi}\right) \tag{12}$$

where $f(\underline{x}_i)$ is the objective value of the $i^{th}$ ant , $\psi, \varphi$ is 1 or 2 and $\mu > 0$. By driving $\mu$ to zero, we penalize the constraint violations with increasing severity.

**Step 3: Pheromone Update**

After evaluating each ant according to $F(\underline{x})$, the pheromone $\tau_{ij}$ on each variable $x_{ij}, i = 1,2,..,n$ of the ant $j, j = 1,2,..,m$ is update according to the evolution step. Pheromone is updated as usually in Equations (13) and (14): first, pheromone is reduced by a constant factor to simulate evaporation to prevent premature convergence; then some pheromone is laid on components of the candidate groups. Accordingly, pheromone concentration associated with each possible route (variable value) is changed in a way to reinforce good solutions, as in Equations:

$$\tau_{ij}(t+1) = (1-\rho).\tau_{ij}(t) + \nabla\tau_{ij}(t), \tag{13}$$

$$\nabla\tau_{ij}(t) = \begin{cases} Q/F(\underline{x}) & \forall x_{ij} \in candidate\ \ groups \\ 0 & otherwise \end{cases} \tag{14}$$

where $Q$ is a constant parameter , $\tau_{ij}(t+1)$ is the revised concentration of pheromone is associated with option $x_{ij}$ at iteration $t+1$, $\tau_{ij}(t)$ is the concentration of pheromone at the previous iteration $t$ and $\nabla\tau_{ij}(t)$ is change in pheromone concentration .

**Step 4: Solution Construction**

Once the pheromone is updated after an iteration, the next iteration starts by changing the ants' paths (i.e. associated variable values) in a manner that respects pheromone concentration as in the Figure (4).

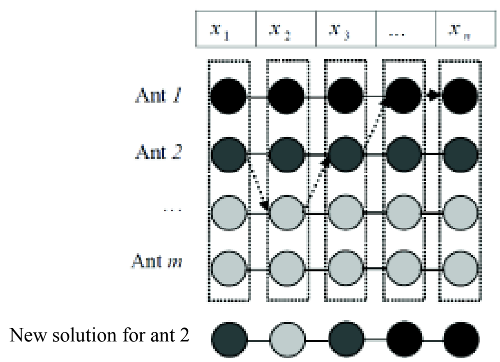


**Fig. (4). Solution construction of ACO.**

For each ant and for each dimension construct a new candidate groups to replace the old one. As such, an ant $a$ chooses the $j^{th}$ value in the group of candidate values for the variable $i$ at iteration $t$ by the roulette wheel selection according to the transition probability as expressed in the Equation (15):

$$p_{ij}^{a}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}}{\sum\limits_{l \in allowed_a}[\tau_{il}(t)]^{\alpha}} & \forall x_{ij}, x_{il} \in allowed_{a} \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

where $p_{ij}^{a}(t)$ is probability that option $x_{ij}$ is chosen by ant $a$ for variable $i$ at iteration $t$ and $allowed_a$ is the set of the candidate values contained in the group.

**Step 5: FFA-Based Search**

The step begins by generating a number of fireflies, $m$ in parallel with the ACO algorithm and the resulting solutions of ACO and FFA are gathered and then evaluated according to the degree of constraint violation based on feasibility rule that determined the reflected back solution into the two populations at the end of iterations. The main steps of the FFA are summarized as follows:

**Step 5.1: Initialization**

Initialize a swarm of fireflies, where each firefly contains $n$ variables (i.e., the position of the $ith$ firefly in the $n$-dimensional search space can be represented as $\underline{x}_i, \underline{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,n})$).

**Step 5.2: Variation of Light Intensity**

Calculate the light intensity $I(\underline{x}_i)$ for each firefly which in turn is associated with the $F(\underline{x}_i)$. Here the brightness $I$ of a firefly is selected to reflect its current position $\underline{x}$ of its fitness value $F(\underline{x})$, given by Equation (16):

$$I(\underline{x}_i) = F(\underline{x}_i) \quad , i = 1,2,\ldots,m \tag{16}$$

**Step 5.3: Movement Toward Attractive Firefly**

The movement of a firefly is formulated as mentioned earlier in the Equation (10) but we present an improvement on the convergence of the algorithm by varying the randomization parameter $\alpha_1$ so that it decreases gradually as the optima are approaching by using the Equation (17):

$$\alpha_{t+1} = \alpha_t \, \theta^{(1-\frac{t}{T})}, t = 1,2,\ldots,T \tag{17}$$

where $T$ is the maximum number of generations for FFA, and $\theta \in (0,1)$ is the randomness reduction constant.

**Step 6: Feasibility Rule**

After ACO and FFA 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, 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 for ACO and the other for FFA. The idea of feasibility rule is based to three criteria: (1) any feasible solution is preferred to any infeasible solution; (2) between two feasible solutions, the one with better objective function value is preferred; (3) between two infeasible solutions, the one with smaller degree of constraint violation is preferred, the reader is referred to Deb [16].

If the variable value $x_{ij}$ of $\underline{x}_i$ violates the boundary constraint, violating variable value is reflected back from the violated boundary using the Equation (18):

$$x_{ij} = \begin{cases} x_{ij}^{L} & \text{if } x_{ij} < x_{ij}^{L} \\ x_{ij}^{U} & \text{if } x_{ij} > x_{ij}^{L} \end{cases} \tag{18}$$

The flow chart which describe the working of ACO-FFA shown in Figure (5) and the basic steps of the proposed ACO-FFA approach can be summarized as the pseudo code shown in Figure (6).

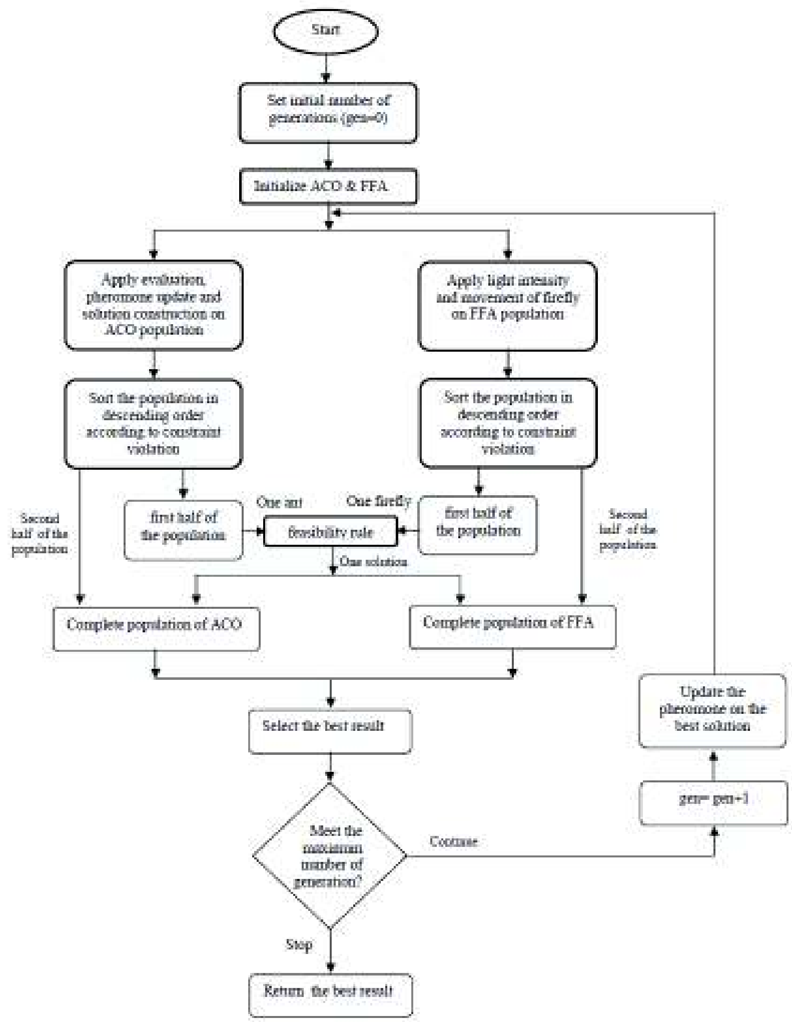


Fig.(5). The flow chart of ACO-FFA.

**Input:** Population size for ACO and FFA, parameters for ACO, parameters for FFA, objective function, constraints, upper bound of variables $x_i^U$ and lower bound of variables $x_i^L, i = 1,2,...,n.$

**Output:** The best objective function value.

$t = 0$

Generate initial population of ants $\underline{x}_a, a = 1,2,...,m(pop_1^t)$, randomly.

Initialize the quantity of pheromone for every value in the candidate groups ( $\tau_0$ ).

Generate initial population of fireflies $\underline{x}_a, a = 1,2,...,m(pop_2^t)$, randomly.

**for** each generation **do**

    **while** $t <= \text{maxiteration}$ **do**

        Evaluate $pop_1^t$ according to Equations (11) and (12).

        Update the pheromone using Equations (13) and (14).

        $pop_1^{t+1}$ =Construct new solution according to Equation (15).

        Modify its variables by Equation (18).

        $t = t + 1$ .

    **end while**

    **while** $t <= \text{maxiteration}$ **do**

        Calculate the light intensity for $pop_2^t$ by Equation (16).

        $pop_2^{t+1}$ = Movement toward attractive firefly by Equation (10).

        Modify its variables by Equation (18).

        $t = t + 1$ .

    **end while**

    $pop_1$ = Sort $pop_1^t$ in descending order according to constraints violation**.**

    $pop_2$ =Sort $pop_2^t$ in descending order according to constraints violation**.**

    $P_1$ = $pop_1$ 's first half part.

    $P_2$ = $pop_2$ 's first half part.

    Maintain the second half of the populations..

    each individual of $P_1$ (denoted as *a*).

    each individual of $P_2$ (denoted as *b*).

    **for** each individual of $P_1$ **do**

        Compare *a* against the corresponding *b* according to feasibility- rule.

        if *a* wins, it replaces the corresponding solution *b* of $P_2$ .

        if *b* wins, it replaces the corresponding solution *a* of $P_1$ .

        if *a* and *b* are not win , the one with smaller degree of constraint violation is preferred.

    **end for**

      $f_{best}$ = the objective function value of the best solutions of $pop_1$ and $pop_2$

    $pop_1^t = pop_1$

    $pop_2^t = pop_2$

**end for**

return $f_{best}$

**Fig. (6). ACO-FFA: A hybrid between ACO and FFA.**

## 5. 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-FFA. The selected problems have been well studied before as benchmarks by various approaches. The main characteristics of 10 benchmark problems (i.e., P1, P2, P3, P4, P5, P6, P7, P8, P9 and P10) are reported in table as shown in Table (1). 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-FFA approach. The algorithm is coded in MATLAB 7and is run on a Pentium IV 2.4 GHz processor with 1.0 GB RAM.

**Table (1). The list of test problems.**

| Test problems | Objective function and constraints | Domain |
|---|---|---|
| P1 | $\min f_1(\underline{x}) = (x_1-2)^2 + (x_2-1)^2$ <br> *subject to* : <br> $g_1(\underline{x}) = x_1 - 2x_2 + 1 = 0$ <br> $g_2(\underline{x}) = x_1^2/4 + x_2^2 - 1 \le 0$ <br> $-10 \le x_i \le 10\ , i = 1,2,...,7.$ | 2 |
| P2 | $\min f_2(\underline{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$ <br> *subject to* : <br> $g_1(\underline{x}) = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.002205x_3x_5$ <br> $g_2(\underline{x}) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2$ <br> $g_3(\underline{x}) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.00190853x_3x_4$ <br> $0 \le g_1(\underline{x}) \le 92,\ 90 \le g_2(\underline{x}) \le 110, 20 \le g_3(\underline{x}) \le 25,$ <br> $78 \le x_1 \le 102,\ 33 \le x_2 \le 45, 27 \le x_i \le 45,\ , i = 3,4,5.$ | 5 |
| P3 | $\min f_3(\underline{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$ <br> *subject to* : <br> $g_1(\underline{x}) = 85.334407 + 0.0056858x_2x_5 + 0.00026x_1x_4 - 0.002205x_3x_5$ <br> $g_2(\underline{x}) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2$ <br> $g_3(\underline{x}) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.00190853x_3x_4$ <br> $0 \le g_1(\underline{x}) \le 92,\ 90 \le g_2(\underline{x}) \le 110, 20 \le g_3(\underline{x}) \le 25,$ <br> $78 \le x_1 \le 102,\ 33 \le x_2 \le 45, 27 \le x_i \le 45,\ , i = 3,4,5.$ | 5 |
| P4 | $\min f_4(\underline{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1$ <br> *subject to* : <br> $g_1(\underline{x}) = 85.334407 + 0.0056858x_2x_5 + 0.00026x_1x_4 - 0.002205x_3x_5$ <br> $g_2(\underline{x}) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2$ <br> $g_3(\underline{x}) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.00190853x_3x_4$ <br> $0 \le g_1(\underline{x}) \le 92,\ 90 \le g_2(\underline{x}) \le 110, 20 \le g_3(\underline{x}) \le 25,$ <br> $78 \le x_1 \le 102,\ 33 \le x_2 \le 45, 27 \le x_i \le 45,\ , i = 3,4,5.$ | 5 |

**Continue table (1).**

| | | |
|---|---|---|
| P5 | $\min f_5(\underline{x}) = (x_1-10)^2 + 5(x_2-12)^2 + x_3^4 + 3(x_4-11)^2 + 10x_5^6$ $+ 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$ *subject to*: $g_1(\underline{x}) = 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 - 127 \le 0$ $g_2(\underline{x}) = 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 - 282 \le 0$ $g_3(\underline{x}) = 23x_1 + x_2^2 + 6x_6^2 - 8x_7 - 196 \le 0$ $g_4(\underline{x}) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \le 0$ $-10 \le x_i \le 10, i = 1,2,\ldots,7.$ | 7 |
| P6 | $\max f_6(\underline{x}) = 0.1(100 - (x_1-5)^2 - (x_2-5)^2 - (x_3-5)^2)$ *subject to*: $g_1(\underline{x}) = (x_1-p)^2 - (x_2-q)^2 - (x_3-r)^2 - 0.0625 \le 0$ $0 \le x_i \le 10, i = 1,2,3.\ p,q,r = 1,2,\ldots,9.$ | 3 |
| P7 | $\max f_7(\underline{x}) = -x_1^2 - x_2^2 - 2x_3^2 - x_4^2 + 5x_1 + 5x_2 + 21x_3 - 7x_4$ *subject to*: $g_1(\underline{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_1 - x_2 + x_3 - x_4 - 8 \le 0$ $g_2(\underline{x}) = x_1^2 + 2x_2^2 + x_3^2 + 2x_4^2 - x_1 - x_4 - 10 \le 0$ $g_3(\underline{x}) = 2x_1^2 + x_2^2 + x_3^2 + 2x_1 - x_2 - x_4 - 5 = 0$ $-100 \le x_i \le 100, i = 1,2,\ldots,5.$ | 5 |
| P8 | $\min f_8(\underline{x}) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3-10)^2 + 4(x_4-5)^2$ $+ (x_5-3)^2 + 2(x_6-1)^2 + 5x_7^2 + 7(x_8-11)^2 + 2(x_9-10)^2$ $+ (x_{10}-7)^2 + 45$ *subject to*: $g_1(\underline{x}) = -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \le 0$ $g_2(\underline{x}) = 10x_1 - 8x_2 - 17x_7 + 2x_8 \le 0$ $g_3(\underline{x}) = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \le 0$ $g_4(\underline{x}) = 3(x_1-2)^2 + 4(x_2-3)^2 + 2x_3^2 - 7x_4 - 120 \le 0$ $g_5(\underline{x}) = 5x_1^2 + 8x_2 + (x_3-6)^2 - 2x_4 - 40 \le 0$ $g_6(\underline{x}) = x_1^2 + 2(x_2-2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \le 0$ $g_7(\underline{x}) = 0.5(x_1-8)^2 + 2(x_2-4)^2 + 3x_5^2 - x_6 - 30 \le 0$ $g_8(\underline{x}) = -3x_1 + 6x_2 + 12(x_9-8)^2 - 7x_{10} \le 0$ $-10 \le x_i \le 10, i = 1,2,\ldots,10.$ | 10 |

**Continue table (1).**

| | | |
|---|---|---|
| P9 | $\min f_9(\underline{x}) = x_1 + x_2 + x_3$ <br> *subject to* : <br> $g_1(\underline{x}) = -1 + 0.0025(x_4 + x_6) \le 0$ <br> $g_2(\underline{x}) = -1 + 0.0025(x_5 + x_7 - x_4) \le 0$ <br> $g_3(\underline{x}) = -1 + 0.01(x_8 - x_5) \le 0$ <br> $g_4(\underline{x}) = -x_1 x_6 + 833.33252 x_4 + 100 x_1 - 83333.333 \le 0$ <br> $g_5(\underline{x}) = -x_2 x_7 + 1250 x_5 + x_2 x_4 - 1250 x_4 \le 0$ <br> $g_6(\underline{x}) = -x_3 x_8 + x_3 x_5 - 2500 x_5 + 1250000 \le 0$ <br> $100 \le x_1 \le 10000$ , $1000 \le x_2, x_3 \le 10000$ <br> $10 \le x_i \le 1000 \quad i = 4,5,...,8.$ | 8 |
| P10 | $\max f_{10}(\underline{x}) = 653.45 x_1 x_4 (0.02825 + x_4 + x_2/2)/(x_4 + 0.00117)$ <br> *subject to* : <br> $g_1(\underline{x}) = 1.92*10^{-5} x_1^2 (0.02825 + x_4 + x_2/2) \big/ x_2 x_3 - -4000 \le 0$ <br> $g_2(\underline{x}) = x_4 + x_2 - 3x_3 - 0.0266 \le 0$ <br> $g_3(\underline{x}) = 1.3 x_4/(0.00117 + x_4) + 165.13 x_3 - 1.5 \le 0$ <br> $g_4(\underline{x}) = 6.28*10^{-7} x_1 (0.00105 + x_4) \big/ x_4 (x_4 + 0.001) - 1.17 \le 0$ <br> $g_5(\underline{x}) = 15000 - 653.45 x_1 x_4 (0.02825 + x_4 + x_2/2)/(x_4 + 0.00117) \le 0$ <br> $0 \le x_1 \le 1028A,\ 0 \le x_2, x_4 \le 0.08m, 0 \le x_3 \le 0.009m.$ | 4 |

## 5.1. 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 (2).

**Table (2). The best parameters for the proposed approach**

| | |
|---|---|
| The size of ant colony ( $m$ ) | 200 |
| Initial pheromone ( $\tau_0$ ) | 100 |
| Evaporation rate ( $\rho$ ) | 0.9 |
| The constant of ACO ( $Q$ ) | 400 |
| Pheromone weight ( $\alpha$ ) | 1 |
| Initial light intensity ( $I_0$ ) | 0 |
| Initial attractiveness ( $\beta_0$ ) | 1 |
| The light absorption coefficient ( $\gamma$ ) | 1 |
| The randomness reduction constant ( $\theta$ ) | 0.9 |
| Randomization parameter ( $\alpha_1$ ) | 0.2 |

### 5.2. Results

The comparison between the results determined by the proposed approach and the compared algorithms are reported in Table (3). The results have demonstrated the superiority of the proposed approach to finding the global optimum solution. Hence the optimum solution for problems P2, P3, P4, P7 and P10 didn't report previously by the literature, we will give the solution for these problems by our approach. Therefore, the solutions obtained by our approach represent great contribution for finding the optimum solutions of these problems.

**Table (3). The comparison of solution quality**

| Test problem | Optimum solution | The obtained solution | | |
|---|---|---|---|---|
| | | The proposed approach | Compared algorithms | |
| | | | Name | Objective value |
| **P1** | 1.3935 | 1.3982800 | Homaifar *et al.* [18]<br>Fogel [19] | 1.4339<br>1.3772 |
| **P2** | - | -30665.544 | Deb [16]<br>Lee and Geem [20] | -30665.500<br>-30665.500 |
| **P3** | - | -31026 | Fasanghary *et al.* [21]<br>Shi and Eberhart [22] | -31024.316<br>-31025.56142 |
| **P4** | - | 6857.7 | Osman et. al [23] | 6857.8 |
| **P5** | 680.6300573 | 680.6395 | Michalewicz [3]<br>Lee and Geem [20] | 680.642<br>680.6413574 |
| **P6** | 1 | 1 | Coello [26]<br>Koziel et. al [24]<br>Mahdavi *et al.* [25] | 1<br>0.999999857<br>0.9999999 |
| **P7** | - | 44.7924 | Fung *et al.* [27]<br>Haralambos *et al.* [28] | 43.98178<br>44.0000 |
| **P8** | 24.3062091 | 24.3377 | Michalewicz [3]<br>Lee and Geem [20] | 24.690<br>24.3667946 |
| **P9** | 7049.3307 | 7053.8 | Michalewicz [3]<br>Lee and Geem [20] | 7377.976<br>7057.274414 |
| **P10** | - | 16236 | Deshpande [29] | 16192 |

The first constrained problem was originally introduced by Bracken and McCormick [17].The optimal solution is $\underline{x} = (0.82288, 0.91144)$ and corresponding optimum solution is $f(\underline{x}) = 1.3935$. On the other hand the optimal solution found using the ACO-FFA is $\underline{x} = (0.8208923, 0.9106394)$ and corresponding optimum solution is $f(\underline{x}) = 1.3982800$. It is clear that the optimum value found using the ACO-FFA is better than the obtained by Homaifar *et al.* [18] and Fogel [19].

The second problem is a minimization problem with five design variables and three inequality constraints bounded from both sides. Deb [16], Lee and Geem [20] didn't mention the optimum solution. Therefore, the obtained solution by the ACO-FFA algorithm is superior to both algorithms, where the optimal solution found by ACO-FFA is $\underline{x} = (78.0, 33.0, 29.99526346, 45.0, 36.77579651)$ and corresponding optimum solution is $f(\underline{x}) = \text{-}30665.544$.

The third problem represents another variation of the second problem which is different in first constraint, has been solved by Fasanghary *et al.* [21] and Shi and Eberhart [22]. The optimum solution by Fasanghary *et al.* [21] and Shi and Eberhart [22] didn't mention. Consequently, the optimal solution of the problem obtained by our approach is $\underline{x} = (78.0, 33.0001, 27.0718, 45.0, 44.9669)$ and corresponding optimum solution is $f(\underline{x}) = \text{-}31026$ .

The fourth problem represents another variation of the third problem which is different in the objective function, has been solved by Osman *et. al*. [23]. Table (3) lists the optimal solution obtained by the ACO-FFA approach is $\underline{x} = ($78.0, 33.0, 27.0718, 45.0, 44.9669) and corresponding optimum solution is $f(\underline{x}) = 6857.7$ , which is better than the solution obtained by Osman *et. al*. [23].

For the fifth problem the optimum solution obtained by the ACO-FFA approach is near to the global optimum solution and superior to Michalewicz [3] and Lee and Geem [20]. The ACO-FFA is also used to solve this problem, and the optimal solution is obtained at $\underline{x} =$ (2.3339, 1.9510, -0.4613, 4.3665,-0.6302, 1.0098, 1.5823) with a corresponding optimum function value equal to $f(\underline{x}) =$ 680.6395. Therefore, the ACO-FFA solution is better than the Michalewicz [3] and Geem [20].

The sixth constrained problem is a maximization problem with three design variables and one inequality constraint. The feasible region of the search space consists of $9^3$ disjointed spheres. A point $(x_1, x_2, x_3)$ is feasible if and only if there exist $p, q, r$ such that the above inequality holds. The optimal is located at $\underline{x} =$ (5, 5, 5) where $f(\underline{x}) =$ 1. The solution lies within the feasible region. The optimum solution obtained by the proposed approach is better than those obtained by Koziel and Michalewicz [24] and Mahdavi *et al.* [25], and it is comparable to the result obtained by Coello [26].

The seventh constrained problem is a maximization problem with five design variables and three inequality constraints. According to Table (3), the optimum solution obtained by our approach is better than those obtained by Fung *et al.*[27] and Haralambos *et al.* [28] , where the optimal solution is obtained at $\underline{x} =$ (0.2492, 1.0415, 2.0611, -0.7520, -19.3181) with a corresponding optimum function value equal to $f(\underline{x}) =$ 44.7924.

The eighth constrained problem is a minimization problem with ten design variables and eight inequality constraints. The optimal solution is $\underline{x} = ($2.171996, 2.363683, 8.773962, 5.095984, 0.9906548, 1.430574, 1.321644, 9.828762, 8.280092, 8.375927) and corresponding optimum function value $f(\underline{x}) = 24.3062091$. Table (3) demonstrates the obtained solution using the ACO-FFA is better than those obtained by Michalewicz [3] and Lee and Geem [20]. Therefore the optimal solution is obtained at $\underline{x} =$ (1.6648, 2.5571, 7.6574, 5.3184, 0.4156, 1.1132, 0.9386, 9.8087, 9.7643, 7.0051) with a corresponding optimum function value equal to $f(\underline{x}) =$ 24.3377.

The ninth problem is a minimization problem with eight design variables and six inequality constraints. The optimum solution is $\underline{x} =$ (579.3167, 1359.943, 5110.071, 182.0174, 295.5985, 217.9799, 286.4162, 395.5979) where $f(\underline{x}) =$ 7049.3307. The ACO-FFA is applied to this problem and the optimal solution found by the ACO-FFA is $\underline{x} =$ (1453.4, 2064, 3536.5, 195.1, 103.7, 172.3, 465.5, 689.0), and its corresponding objective function value is equal to $f(\underline{x}) = 7053.8$. Table (3) lists the comparisons of results, and it is clear that the best objective function value obtained using the ACO-FFA is superior to those reported previously in the literature.

The tenth constrained problem is the linear synchronous motor design. The linear synchronous motor structure (as in Fig. 7) is a practical design problem that has been often used as a benchmark for testing different optimization methods [29]. The objective is to find the maximum force generated by the linear synchronous motor subject to constraints on heat, radius, saturation, demagnetization and maximum force constraint. There are four design variables: the current in each slot, $ins(x_1)$ the dimensions of the slot, $h_s(x_2)$ and $t_s(x_3)$ , the height of the magnet, $h_m(x_4)$. The mathematical formulation of the objective function tenth constrained problem is as follows:

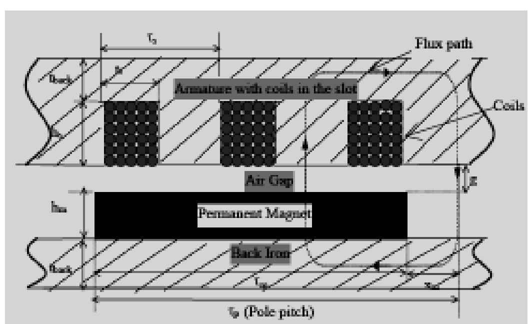


**Fig. (7). Cross-section of approximately a one pole pitch long section of linear synchronous motor**

The comparisons of results are shown in Table (3). The ACO-FFA result is superior to those obtained using the Deshpande [29]. Therefore the objective function value obtained by our method is better than those obtained by Deshpande [29], where the optimal solution found by the ACO-FFA is $\underline{x} =$ (368.4769, 1.1, 32. 8, 26 .1), and its corresponding objective function value is equal to $f(\underline{x}) = 16236$.

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 approach has been used to increase the solution quality by combining the two merits of two heuristic algorithms. On the other hand, unlike classical techniques our approach search from a population of points, not single point. Therefore our approach can provide a globally optimal solution. 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-FFA 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.

## 6. Conclusions

This paper presents a hybrid approach combining two heuristic optimization techniques, ACO and FFA. Our approach integrates the merits of both ACO and FFA to solve COPs, where ACO-FFA is applied with two kinds of swarm for evolutionary exploration and exploitation in spaces of both solutions and penalty factors. In particular, one kind of a single swarm is used the ants and another kind of a single swarm is used the fireflies which are used in parallel to search good decision solutions. 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-FFA is tested on several benchmark problems from the usual literature. From the comparative study, ACO-FFA has shown its potential to handle various COPs, and its performance is much better than other state-of-the-art. A careful observation will reveal the following benefits of the proposed optimization technique.

1.It can efficiently overcome the drawback of classical ant colony algorithm which is not suitable for solving constrained optimizations.

2.It competitive when compared with the other existing algorithm.

3.It can find can find the global optimum solution for the problems very efficiently.

4.The candidate paths to be selected by the ant changes dynamically rather than to be fixed and the solutions will tend to be diverse and global by means of using firefly algorithm on the component values of the population at each iteration.

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

## 7. References

[1] C. Blum, A. Roli , Metaheuristics in combinatorial optimization: overview and conceptual comparison. ACM Comput. Surveys, 35(3):268–308, 2003.

[2] F. Glover, G. Kochenberger, editors. Handbook of metaheuristics. Nonvell, 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 Kauffman, San Mateo, 151–158, 1995.

[4] M. Yagiura, T. Ibaraki, On metaheuristic algorithms for combinatorial optimization problems. Systems and Computers in Japan;32(3):33–55, 2001.

[5] M. Dorigo, Learning and Nature Algorithm (in Italian). Ph.D. Dissertation, Dipartimento di Electtonica, Politecnico di Milano, Italy, 1992.

[6] M. Dorigo, T. Stützle, Ant Colony Optimization, MIT Press, London, 2004.

[7] L. Chen, J. Shen, L. Qin , H J. Chen, An improved ant colony algorithm in continuous optimization. Journal of Systems Science and Systems Engineering, 12(2), 224-235, 2003.

[8] J. Dreo, P. Siarry, An ant colony algorithm aimed at dynamic continuous optimization. Applied Mathematics and Computation, 181, 457-467, 2006.

[9] A. A. Mousa, Waiel 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.

[10] P. Hadi, T.M. Reza, Solving a multi-objective open shop scheduling problem by a novel hybrid ant colony optimization, Expert Systems with Applications 38, 2817–2822,2011.

[11] Y. Jingan, Z. Yanbin, An improved ant colony optimization algorithm for solving a complex combinatorial optimization problem, Applied Soft Computing 10,pp. 653–660,2010.

[12] Z. Xiaoxia, T. Lixin, A new hybrid ant colony optimization algorithm for the vehicle routing problem, Pattern Recognition Letters 30 ,848–855, 2009.

[13] X.S. Yang.: Nature-Inspired Metaheuristic Algorithms. Luniver Press ,2008.

[14] M. Dorigo, E. Bonaneau, , G. Theraulaz, Ant algorithms and stigmergy. Future Generation Computer Systems, 16, 851–871,2000.

[15] 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.

[16] K. Deb, An efficient constraint handling method for genetic algorithms, Computer Methods in Applied Mechanics and Engineering,186 :311 -338, 2000.

[17] J. Bracken, GP. McCormick , Selected applications of nonlinear programming. New York: John Wiley and Sons; 1968.

[18] A. Homaifar, S. H. V. Lai, X. Qi, Constrained optimization via genetic algorithms. Simulation,62(4):242–54, 1994.

[19] DB. Fogel , A comparison of evolutionary programming and genetic algorithms on selected constrained optimization problems. Simulation, 64(6):399–406, 1995.

[20] K. S. Lee, Z. W. Geem, A new meta-heuristic algorithm for continues engineering optimization: harmony search theory and practice. Computer Methods in Applied Mechanics and Engineering;194:3902–33, 2004.

[21] M. Fesanghary, M. Mahdavi, M. Minary-Jolandan, Y. Alizadeh, Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems: harmony search theory and practice. Computer Methods in Applied Mechanics and Engineering, 197:3080–91, 2008.

[22] Y. Shi, R. C. Eberhart , A modified particle swarm optimizer. In: Proceedings of the international congress on evolutionary computation 1998. Piscataway, NJ: IEEE Service Center; 69–73, 1998.

[23] M.S. Osman , Mahmoud A. Abo-Sinna, A.A. Mousa, A Combined Genetic Algorithm-Fuzzy Logic Controller (GA-FlC) in Nonlinear Programming. Applied Mathematics and Computation 170, 821–840, 2005.

[24] S. Koziel, Z. Michalewicz, Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization, Evolutionary Computation 7 (1) ,19–44, 1999.

[25] M. Mahdavi, M. Fesanghary, E. Damangir, An improved harmony search algorithm for solving optimization problems, Applied Mathematics and Computation 188 (2) ,1567–1579, 2007.

[26] C.A.C. Coello, Constraint-handling in genetic algorithms through the use of dominance-based tournament selection, Advanced Engineering Informatics 16 (3) 193–203, 2002.

[27] R. Fung, J. Tang, D. Wang, Extension of a hybrid genetic algorithm for nonlinear programming problems with equality and inequality constraints. Computers and Operations Research, 29:261–74, 2002.

[28] S. Haralambos , N. Athanassios, A line up evolutionary algorithm for solving nonlinear constrained optimization problems, Computers & Operations Research, 32 , 1499–1514, 2005.

[29] A. D. Deshpande, California Linear Drives, Inc., www.calinear.com, A Study Of Methods To Identify Constraint Dominance In Engineering Design Problems, M.SC. Thesis, Mechanical and Industrial Engineering Department, University of Massachusetts, Amherst, 2002.

# رواية تهجين أمثليه مستعمرة نملة وخوارزم اليراع لحل مشاكل التصميم الهندسية المقيدة

أحمد أحمد الصاوي[1] و السيد محمد محمد زكي[2] و رزق مسعود رزق الله [2]

[1] قسم الرياضيات– كلية العلوم– جامعة القصيم– السعودية

[2] قسم العلوم الأساسية الهندسية– كلية الهندسة بشبين الكوم – جامعة المنوفية - مصر

(قدم للنشر ٢٠١٢/٦/٢٤ ، قبل النشر١ /٢٠١٢/٩)

**ملخص البحث**. هذه الورقة تعرض رواية تهجين أمثليه مستعمرة نملة وخوارزم اليراع لمشاكل الامثلية المقيدة الذي يدمج المفاهيم من أمثلية مستعمرةِ نملة وخوارزم اليراع . المنهجية للخوار زم المقترح قدمت مستنده علي إلية متوازية لأمثليه مستعمرةِ نملة و خوارزم اليراع لتحديث الحلول للخوار زم المقترح .علاوة علي ذلك تطور هذه الحلول يؤدى بواسطة استخدام قاعدة المناسب التي تبقي الحلول بدرجات منخفضة لانتهاكات القيد . الخوارزم المقترح اختبار علي عديد من الدوال القياسية. المقارنات عرضت أن الخوار زم المقترح يفوق أو يؤدي بطريقه مشابه أحدث المقترحات من ناحية جودة الحلول الناتجة.