A Modified Ant Colony Optimization Algorithm Modeled on Tabu-Search Methods

S. L. Ho[1], Shiyou Yang[2], Guangzheng Ni[2], and Josè Márcio Machado[3]

[1]Department of Electrical Engineering, the Hong Kong Polytechnic University, Hong Kong
[2]College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China
[3]São Paulo State University (UNESP)—São Josà do Rio Preto, Brazil

An earlier model underlying the foraging strategy of a pachycodyla apicalis ant is modified. The proposed algorithm incorporates key features of the tabu-search method in the development of a relatively simple but robust global ant colony optimization algorithm. Numerical results are reported to validate and demonstrate the feasibility and effectiveness of the proposed algorithm in solving electromagnetic (EM) design problems.

***Index Terms*—Ant colony optimization (ACO), design optimization, foraging behavior, tabu search method.**

I. Introduction

ANT COLONY optimization (ACO) is an evolutionary algorithm (EA) to model the behavior of almost blind ants in establishing the shortest path from their colony to their feeding sources and back [1], [2]. As an ant moves, it lays varying amount of pheromones, which are detectable by other ants, along its path, thereby marking the path by a trail of such substances. As more ants pass by, more pheromones are deposited on the path. Because ants chase after pheromones, the richer the trail of pheromones in a path is, the more likely it would be followed by other ants. Hence, ants can establish the shortest way from their colony to the feeding sources and back. Moreover, the collective behavior of ant colonies exhibits the so called "autocatalytic" characteristics [3]. By virtue of the learning and searching power of an autocatalytic self-organization such as an ant colony, ACO-based algorithms are receiving increasing attention and have enjoyed great success in the solution of traditionally difficult optimization problems [1]–[4].

To date, most of these algorithms are confined to applications on combinatorial optimizations, and there are few applications to problems with continuous variables [3], [5], [6]. In addition, the common feature of the available continuous ACO algorithms is that the search process is divided into bilevel search procedures (i.e., a local and a global search one); and two genetic algorithms (GAs) are used in the global search procedure. Because of the complexity as described, the search procedure is very complex in terms of numerical implementation and parameter tuning. In this regard, the authors have successfully proposed an improved continuous ACO algorithm with an emphasis on the design of a simple, novel, but robust global search mechanism in solving typical electromagnetic (EM) design problems [7]. However, even the aforementioned approach finds the best solution in simulated cases, the search procedure is still not simple in terms of numerical implementations. For example, the tedious pheromone updating scheme is explicitly inherited into such an algorithm. Hence, it is proposed that the aforementioned model underlying the foraging strategy of the pachycodyla apicalis (API) ant is modified by incorporating key features of the tabu-search methods in the proposed novel ACO algorithm.

Digital Object Identifier 10.1109/TMAG.2006.871425

II. A Modified ACO Algorithm

A. Simple Description of the Foraging Strategy of Pachycondyla Apicalis Ants and the API Algorithm [8]

Pachycondyla apicalis ants have been studied in the Mexican tropical forest near the Guatemalan boarder. Colonies of these ants comprise around 20 to 100 ants. The foraging strategy of such ants can be characterized as follows. First, these ants create their hunting sites which are distributed relatively uniformly around their nest within a radius of approximately 10 m. In this way, using a small mosaic of areas, the ants cover a rather large region around the nest. Second, the ants will intensify their searches around some selected sites for prey. In this foraging process, these ants communicate with each other using visual landmarks rather than pheromone trails. After capturing their prey, the ants will move to a new nest based on a recruitment mechanism called tandem running to begin a new cycle of foraging. Obviously, these ants use relatively simple principles, both locally and globally, in their search for prey. Based on such a metaphor, Monmarche *et al.* propose an API algorithm for the solution of optimization problems [8]. In the API algorithm, the search procedure is also divided into two different ones (i.e., a global one based on modeling the nest movement for exploration, and a local one based on a modeling of the hunting site in their search for extensive exploitations in optimization studies). However, further research, together with our experiments, shows that: 1) the API makes poor use of the memory that generally characterizes ant colony systems [9] and 2) the global search ability of the API algorithm is not robust.

B. Modified ACO Algorithm

To eliminate the aforementioned shortcomings of the API algorithm, a modified ACO algorithm that incorporates some favorable features of tabu searches is proposed. To facilitate the understanding and description, the stepwise procedures of the proposed algorithm are described.

1) *Initialization*: set the algorithm parameters.
2) *Generation of new nest* (exploration): use the tabu-search method to generate a new nest N.
3) *Exploitation*
   3.1) *Intensification search*: For each ant a_i,
   if a_i has less than p hunting sites in its memory, then create a new site in the neighborhood of N and exploit this new site;
   else
   If the previous site exploitation is successful, then exploit the same site again;
   else exploit a probabilistically selected site (among its p sites in memory).
   3.2) *Information sharing*: probabilistically replace a site in the memory of the ant by the best one searched so far in this cycle.
   3.3) *Nest movement*: if the condition for nest movements is satisfied, go to (4); otherwise, go to (3.1).
4) *Termination test*: If the test is passed, stop; otherwise, empty the memories of all ants and then go to (2).

1) Generation of New Nest: After initialization in an API algorithm, only the best solution found since the last nest move has the opportunity to be selected as a new nest to start the next iteration. Therefore, the algorithm has no "climb up property." However, the key success of an optimal algorithm is its "climb up" ability in the search for global solutions. Thus, there is little wonder that the global searching ability of the API algorithm is not robust. To enhance the global search ability of the API algorithm, a new strategy is designed in the proposed algorithm. To generate the new nest, the tabu-search method is modified and used. First, a new feasible solution is generated for each ant in its neighborhood; second, the best one of these solutions is selected as the new nest regardless of whether the objective function value of this solution is larger than that of the current nest. Contrary to available tabu searches, the neighborhood sizes of each ant are set to be equal to half of the dimensions of the decision parameter space in order to guarantee and enhance the diversity of the newly generated nest.

2) Exploitation: The main differences of the proposed and the API algorithms lie in the following aspects.

Initially, each ant a_i checks its memory. If the number of hunting sites in its memory is less than a predefined number p, it will generate a new one in the small neighborhood of the current nest, saves it to its memory, and uses it as a hunting site. Otherwise, one of its memorized sites is selected, using a Roulette wheel selection scheme according to the site's fitness values, as the hunting site.

Ant a_i then performs a local search around the neighborhood of this hunting site. If this local exploitation is successful, ant a_i will repeat its exploration around the site until an unsuccessful search occurs; otherwise, the ant will select an alternative one among its memorized sites using the same Roulette wheel selection scheme to begin the next cycle of exploitations.

This process will be repeated until a termination criterion is satisfied. The termination criterion used in this phase is that the procedure will stop automatically once the number of successive unsuccessful explorations reaches a predefined value. To keep the diversity of hunting sites in this iterative process in order to find the relatively promising solutions, an age variable is assigned to the hunting sites. Once a hunting site is generated, its "age" will be assigned a minimal value, and this value will increase incrementally each time the hunting site is chosen. If the "age" of a member in the memory of an ant reaches a predefined value, the specific site will be erased from the ant's memory.

It should be noted that ants of the proposed algorithm uses visual landmarks, rather than pheromone trails as used by many other ant species, as the communications medium. Thus, a relatively simple procedure, when compared with other ant-based ones, could be developed as reported in this paper.

3) Information Sharing: As described previously, the available API algorithm makes poor use of memories that generally characterizes ant colony systems. On the other hand, communication using landmarks among ants in available API algorithms is not as obvious as those used in other ant methods. Consequently, the information accumulated by the searched solutions is not used fully in API algorithms, thereby degrading their search efficiency. To compensate for such shortcoming in an API algorithm, an information-sharing mechanism is proposed so as to make full use of the information gathered from the latest searched solutions, so as to speed up the solution process. In essence, it is proposed that after each ant has extensively exploited a hunting site, one member of the memorized sites for every ant will be replaced by the best one searched so far by all of the ants in this cycle. For the selection of the replaced sites, the Roulette wheel selection scheme is used.

III. Numerical Examples

The proposed ACO algorithm is experimented extensively on both well-designed mathematical functions and optimal design problems in electrical engineering as reported below.

A. Experiments

The experiments of the proposed ACO algorithm on two mathematical functions selected from [9] are first presented to serve as examples for comparing the performances of the proposed algorithm with those of the available optimal methods. The details about the first function are defined as

$$\begin{aligned} &\min \quad 50 + \sum_{i=1}^{5} \left(x_i^2 - 10\cos(2\pi x_i)\right) \\ &\text{subject to} \quad -5.12 \le x_i \le 5.11 \, (i = 1, 2, \ldots, 5). \end{aligned} \tag{1}$$

The second test function is also a five-dimensional one, and is formulated as

$$\begin{aligned} &\min \quad 1 + \sum_{i=1}^{5} \left(x_i^2/4000\right) - \prod_{i=1}^{5} \cos(x_i/\sqrt{i}) \\ &\text{subject to} \quad -5.12 \le x_i \le 5.11 \, (i = 1, 2, \ldots, 5). \end{aligned} \tag{2}$$

These two functions have a common minimum value of zero at a common point (0,0,0,0,0). In all of the numerical experiments, the parameters of the proposed modified ACO algorithms are set to be the same as those of the API algorithm [9]

TABLE I
PERFORMANCE COMPARISON OF DIFFERENT OPTIMAL ALGORITHMS ON THE FIRST MATHEMATICAL FUNCTION FOR 50 INDEPENDENT RUNS

Algorithms	Objective function			Iteration number		
	Best	Worst	Averaged	Min	Max	Averaged
Proposed	0.00	0.00	0.00	6900	9945	8090
API [a]	4.83	28.79	10.52	10000	10000	10000
SA	0.00	0.00	0.00	9420	12135	10505

[a] The results are quoted (calculated) from Table 2 of [9].

TABLE II
PERFORMANCE COMPARISON OF DIFFERENT OPTIMAL ALGORITHMS ON THE SECOND MATHEMATICAL FUNCTION FOR 50 INDEPENDENT RUNS

Algorithms	Objective function			Iteration number		
	Best	Worst	Averaged	Min	Max	Averaged
Proposed	0.00	0.03	0.01	6945	15135	10565
API [a]	0.09	1.27	0.30	10000	10000	10000
SA	0.00	0.01	0.00	99142	16735	12485

[a] The results are quoted (calculated) from Table 2 of [9].

whenever applicable. The proposed algorithm will stop its iterative procedure automatically if the number of successive iterations without improvements in the objective function reaches 200. For comparative purposes, the two test mathematical functions are solved using the proposed modified ACO, the API, and a standard SA algorithm. For each test function, all three methods are run independently 50 times with randomly generated initial solutions (nests). The minimal, maximal, and averaged performances of different methods for the first and second test functions are given, respectively, in Tables I and II. In these two tables, the results are corrected to four digital numerical precisions. Tye following can be seen from these numerical results.

1) With respect to the global search ability, the proposed one is more advantageous when compared to the original API algorithm, especially for the first test function.
2) In terms of computational efficiency, the proposed one is comparable to that of its precursor, the API algorithm.
3) The original API algorithm is not robust in finding the global optimal solution of a multimodal objective function.
4) Compared with the well-designed SA algorithm, the global searching ability of the proposed algorithm is degraded marginally.
5) The convergence speed of the proposed algorithm is faster than that of the SA algorithm.
6) When compared with the SA algorithms, the proposed one is, however, relatively awkward in terms of computer implementations.

To summarize, the proposed algorithm is the second best in terms of global search ability, and is the best in terms of searching efficiency, among the three optimal methods being examined in the aforementioned experiments.

B. Validation Using Benchmark Problem

After testing the proposed algorithm on mathematical functions, it is then used to solve the TEAM Workshop problem 22 of a superconducting magnetic energy storage (SMES) configuration with three free parameters, as shown in Fig. 1. The optimal design of a SMES configuration can be formulated as

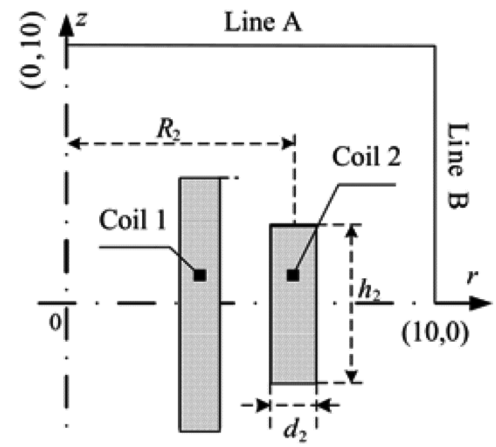

Fig. 1. Schematic diagram of an SMES.

$$\min f = \frac{B_{\text{stray}}^2}{B_{\text{norm}}^2} + |\text{Energy} - E_{\text{ref}}|E_{\text{ref}}$$
$$\text{s.t. } J_i \leq -6.4|(B_{\max})_i| + 56(\text{A/mm}^2) \tag{3}$$

where Energy is the stored energy of the SMES device, $E_{\text{ref}} =$ 180 MJ, $B_{\text{norm}} = 3 \times 10^{-3}$ T, J_i and $(B_{\max})_i (i = 1, 2)$ are, respectively, the current density and the maximum field in the ith coil; and B_{stray}^2 is a measure of the stray fields which is evaluated along 22 equidistant points of Line A and Line B of Fig. 1 using the following expression:

$$B_{\text{stray}}^2 = \sum_{i=1}^{22} (B_{\text{stray}})_i^2 / 22. \tag{4}$$

In this optimal problem, the parameters to be optimized are the geometric parameters of coil 2, as shown in Fig. 1, and the details about the constraints imposed on these parameters are given in [10]. In the numerical implementation, the performances as required for (3) and (4) are determined based on the finite-element solutions. The number of ants is set to be 10; the neighborhood sizes in the exploitation phase are set to be one-tenth of the space dimension. The maximum explorations for a nest is set to be 60 iterations; The proposed algorithm will stop its iterative process when the number of successive iterations without improvements in the objective function reaches 120. Table III gives the computed solutions of the proposed algorithm as well as the best ones searched so far by the Institut für Grundlagen und Theorie der Elektrotechnik (IGTE). Clearly, the proposed algorithm yields nearly identical solutions to the currently best ones contributed by IGTE, making this verification really meaningful.

C. Application

Finally, the proposed ACO algorithm is used to solve the optimal design of a cold crucible (Fig. 2) in which the conductive wall is segmented by longitudinal slits and with each electrically isolated segment being internally or externally cooled by

TABLE III
COMPARISON OF THE COMPUTED AND THE BEST SOLUTIONS FOR SOLVING TEAM WORKSHOP PROBLEM 22 USING THE PROPOSED ALGORITHM

Method	r_2(m)	$h_2/2$ (m)	d_2(m)	Objective	No. iterations
Proposed	3.08	0.242	0.389	10.898×10^{-2}	2468
By IGTE	3.08	0.239	0.394	8.808×10^{-2}	/

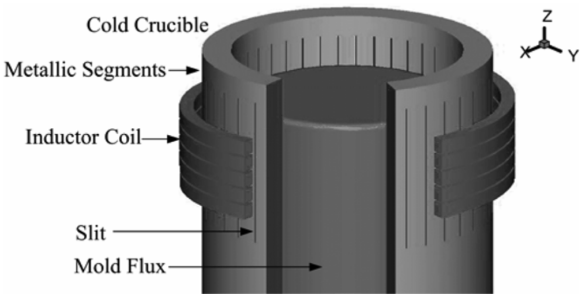

Fig. 2. Schematic diagram of a cold crucible.

water. In the cold crucible, the pinching force resulting from the interaction of the imposed EM field and the induced eddy current in the molten metal is utilized to prevent the latter from making contact with the crucible walls. To achieve this goal, a high-frequency exciting current is applied to the inductor coil. Consequently, the design objective of a crucible is to minimize the Joule losses and ensure the exciting coil is resonating with a capacitor at the operating high frequency of the power supply. The design parameters to be optimized include the frequency of the power supply, the number of turns of the coil, the number of slits, and the outside radius of the crucible. Mathematically, this problem is formulated as

$$\begin{aligned} \min \quad & f = \frac{\mathrm{JPower}}{\mathrm{JPower}_{\mathrm{norm}}} + \frac{|\mathrm{Inductance}-\mathrm{In}_{\mathrm{ref}}|}{\mathrm{In}_{\mathrm{ref}}} \\ \text{s.t.} \quad & B \geq B_{\min} \end{aligned} \tag{5}$$

where JPower is the Joule power losses of the crucible, Inductance is the equivalent inductance of the coil, B is the average magnetic flux density observed on the molten metal surface along the circumferential directions at the center position of the coil, and JPower $_{\mathrm{norm}}$ and In $_{\mathrm{ref}}$ are two references.

The A-V-A finite-element method is used to calculate the eddy current field and to determine the quantities of (5) subsequently. The parameters used for the proposed algorithm are the same as those used for the benchmark problem, and those for the API algorithm are the same as those as reported in [9]. This problem is solved, respectively, using the proposed and the API algorithms. It is found that the proposed algorithm uses 2948 iterations to obtain a solution with a minimum of 1.01 whereas the API algorithm needs 3024 iterations to yield a solution with a minimum of 1.02. The performance comparison of these two methods is given in detail in Table IV. From the numerical results of the proposed algorithm on such a difficult engineering design problem involving sophisticated simulations of high-frequency eddy current fields, it is obvious that the proposed algorithm can find an improved solution while the iteration number used by it is slightly smaller than that used by the original API algorithm.

TABLE IV
PERFORMANCE COMPARISON OF PROPOSED AND API ALGORITHMS

Algorithms	Slit number	Frequency	Out radius (pu)	Turn number	No. of iterations	Objective
Proposed	24	50 k Hz	1.0	3	2948	1.01
API	16	50 k Hz	1.02	3	3024	1.02

IV. CONCLUSION

A modified ACO algorithm is proposed and validated on three different types of problems with promising numerical results, demonstrating its feasibility and robustness in the study of engineering design problems. To the best of our knowledge, there are very few applications of ACO algorithms in the computational electromagnetics community. Therefore, the authors hope that this work will stimulate extensive investigations on the use of ACO algorithms in the design optimization studies of EM devices.

ACKNOWLEDGMENT

This work was supported by the Research Grant Council of the Hong Kong Special Administrative Region, China, under Project PolyU 5242/04E.

REFERENCES

[1] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies," in *Proc. ECAL91-European Conf. Artificial Life*, 1991, pp. 134–142.

[2] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperative agents," *IEEE Trans. Syst., Man, Cybern., B Cybern.*, vol. 26, no. 1, pp. 29–41, Feb. 1996.

[3] G. Bilchev and I. C. Parmee, "The ant colony metaphor for searching continuous design spaces," *Lecture Notes Comput. Sci.*, vol. 993, pp. 25–39, 1995.

[4] C. Gagne, W. L. Price, and M. Gravel, "Comparing an ACO algorithm with other heuristics for the single machine scheduling problem with sequence-dependent setup times," *J. Oper. Res. Soc.*, vol. 53, pp. 895–906, 2002.

[5] M. Wodrich and G. Bilchev, "Cooperative distributed search: The ants' way," *Contr. Cybern.*, vol. 26, pp. 413–446, 1997.

[6] M. Mathur, S. B. Karale, S. Priyee, V. K. Jayaraman, and B. D. Kulkarni, "Ant colony approach to continuous function optimization," *Ind. Eng. Chem, Res.*, vol. 39, pp. 3814–3822, 2000.

[7] S. L. Ho, S. Yang, H. C. Wong, E. K. W. Cheng, and G. Ni, "An improved ant colony optimization algorithm and its application to global optimizations of electromagnetic devices," *IEEE Trans. Magn.*, vol. 41, no. 5, pp. 1764–1767, May 2005.

[8] N. Monmarche, G. Venturini, and M. Slimane, "On how pachcondyla apicalis ants suggest a new search algorithm," *Fut. Gen. Comput. Syst.*, vol. 16, pp. 937–946, 2000.

[9] J. Dreo and P. Siarry, "A new ant colony algorithm using the heterarchical concept aimed at optimization of multiminima continuous functions," *Lecture Notes Comput. Sci.*, vol. 2463, pp. 216–221, 2002.

[10] Team Workshop Problem 22–A Superconducting Magnetic Energy Storage Benchmark [Online]. Available: http://www-igte.tu-graz.ac.at/archive/team/team3dis.htm

Manuscript received June 20, 2005; revised November 20, 2005 (e-mail: eeslho@polyu.edu.hk).