The Multilevel Ant Stigmergy Algorithm: An Industrial Case Study

Peter Korošec, Jurij Šilc

Computer Systems Department, Jožef Stefan Institute, Ljubljana, Slovenia

Abstract

The paper presents a new Multilevel Ant Stigmergy Algorithm for minimizing the losses in a universal electro-motor by optimizing the independent geometrical parameters of the rotor and the stator. We have proposed a general approach for translating a multi-parameter problem into a graph representation. To find a solution (a path in the graph) we used a stigmergic system based on an ant colony. The average solution obtained with the Multilevel Ant Stigmergy Algorithm is 25% better than a solution recently found using a genetic algorithm.

Keywords: stigmergy, ant-colony optimization, multilevel method, multi-parameter problem.

1. Introduction

Stigmergy is a method of communication in decentralized systems in which the individual parts of the system communicate with one another by modifying their local environment. Stigmergy was first observed in nature; for example, ants communicate with one another by laying down pheromones along their trails, so an *ant colony* is a stigmergic system. The term stigmergy (from the Greek *stigma* = sting, and *ergon* = work) was originally defined by the French biologist Pierre-Paul Grassé in his pioneering studies on the reconstruction of termite nests [1]. He defined it as: "Stimulation of workers by the performance they have achieved." It is now also employed in experimental research in robotics, multi-agent systems and communication in computer networks.

In this paper we introduce a new stigmergy-based approach to the multi-parameter optimization problem. In particular, we improve the efficiency of a universal electro-motor, where the goal is to find a new set of independent geometrical parameters for the rotor and the stator with the aim of reducing the motor's power losses, which occur in the iron and the copper.

The rest of the paper is organized as follows. Following the brief outline of the new optimization algorithm in Section 2, the electro-motor power-losses problem is defined in Section 3. In section 4, the empirical results are given. Finally, we conclude the paper in Section 5.

2. The Multilevel Ant Stigmergy Algorithm

First, we translate the multi-parameter problem into a directed graph. Each path on the graph represents one possible solution, and the whole graph represents the whole solution space of the multi-parameter problem. After the translation some sort of optimization technique is used to find the cheapest path in the constructed graph; this path consists of the values of the optimized parameters. In our case we use an optimization algorithm called the *Ant Stigmergy Algorithm* (ASA), the routes of which can be found in the ant colony optimization (ACO) method [2].

We consider the *multilevel paradigm* and its potential to aid the solution of optimization problems. The multilevel paradigm is a simple one, which at its most basic involves recursive coarsening to create a hierarchy of approximations to the original problem. An initial solution is found (sometimes for the original problem, sometimes at the coarsest level) and then iteratively refined at each level. As a general solution strategy the multilevel procedure has been in use for many years and has been applied to many problem areas. However, with the exception of the graph-partitioning problem [3, 4], multilevel techniques have not been widely applied to combinatorial optimization problems [5].

This approach is called the *Multilevel Ant Stigmergy Algorithm* (MASA). The MASA consists of five main phases: graph construction (Phase 1), coarsening (Phase 2), optimization (Phase 3), refinement (Phase 4), and local optimization (Phase 5).

2.1. Graph construction and coarsening

Phase 1 is the initialization, where we translate the parameters of the problem into a directed graph. In this way we translate the multi-parameter problem into a problem of finding the cheapest path. Fig. 1 shows how this is done. We can see that each P_d, $d = 1, \ldots,$

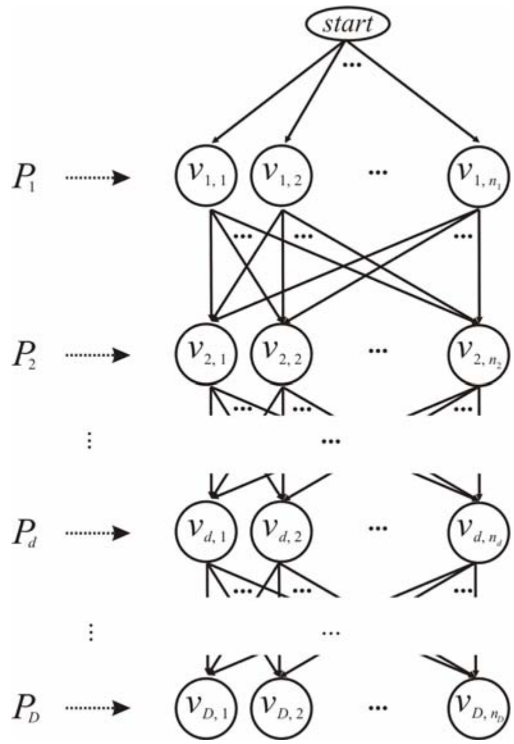

Fig. 1: Phase 1 — Graph construction.

D, parameter value $v_{d,i}$, i = 1, …, n_d, and n_d, = $|P_d|$, represents one vertex in a graph, and each vertex is connected to all the vertices that belong to the next parameter P_{d+1}.

Once we have translated the multi-parameter problem into one of finding the cheapest path, we can proceed with Phase 2. Here we coarsen the graph to some predetermined size. Coarsening is done by merging two or more neighboring vertices into one vertex (Fig. 2); this is done in L iterations (we call them levels). On this coarsened graph we deploy the initial pheromone values on all graph vertices (initialization).

2.2. Optimization

Now we are ready for Phase 3. Here we apply optimization with the ASA. We have m ants that all simultaneously start from the s*tart* vertex. The probability with which they choose the next vertex depends on the amount of pheromone on the vertices (probability rule). Ants repeat this action until they get to the ending vertex. Now we evaluate the gathered parameter values of each ant (that can be found on its path) through the use of some sort of evaluation technique. Now each ant returns to the *start* vertex and on the way it deposits pheromones on the vertices according to the evaluation result: the better the result the more pheromone is deposited on the vertices, and vice versa. If the gathered parameters form an infeasible solution then the amount of pheromones on the parameter vertices is slightly decreased. After all the ants return to the *start* vertex we do a daemon action, which consists of depositing some additional pheromones on currently best path and also some smaller amount on a neighborly path. Afterwards the pheromones are

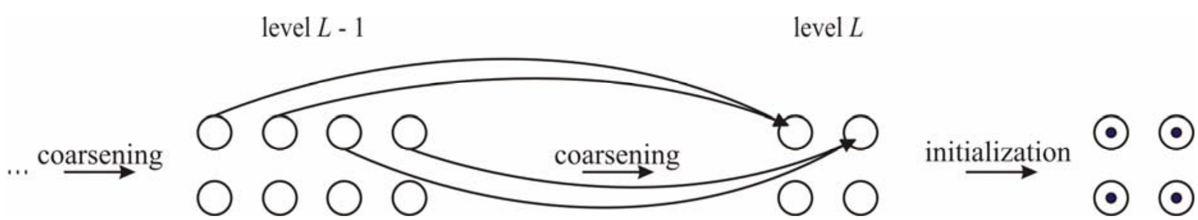

Fig. 2: Phase 2 — Tree coarsening.

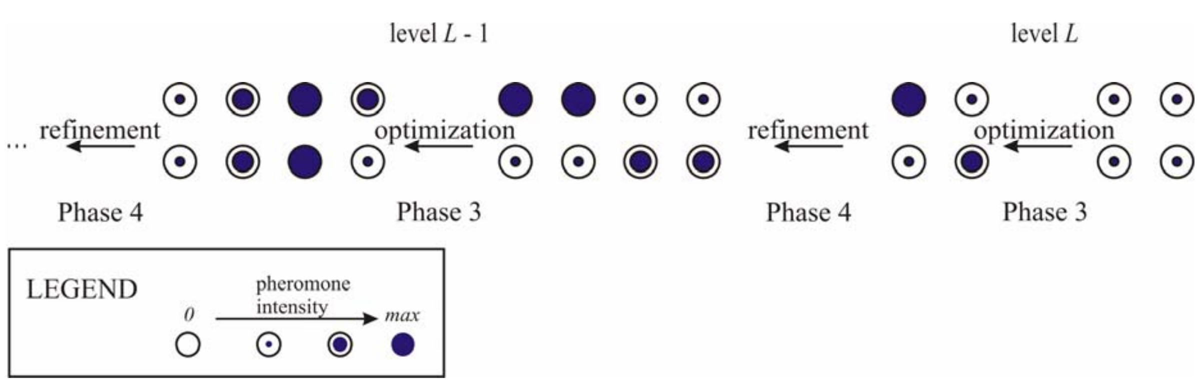

Fig. 3: Phases 3 and 4 — Graph refinement with optimization.

evaporated on all vertices, i.e., the amount of pheromones is decreased by some predetermined percentage on each vertex. The whole procedure is repeated until some ending condition is met.

2.3. Graph refinement

Once the ending condition is met we continue with Phase 4, where we refine the coarsened graph by one level. Because of the simplicity of the coarsening in Phase 2, the interpolation itself is very trivial. All the vertices that are created from one vertex have the same amount of pheromone intensity as that one.

When this is done, we continue with Phase 3. We repeat these two phases until the graph is expanded to its original size and the ASA is run on every level of the expansion (Fig. 3).

2.4. Local optimization

Phase 5 consists of local optimization, which finds a local minimum of a currently best solution. It is a type of steepest descent algorithm, which will efficiently find a local optimum.

2.5. The algorithm

The outline of the MASA pseudo code is as follows:

```
                                                    Phase 1
graph = Initialization(parameters)

                                                    Phase 2
For level = 1 to L do
   Coarsening(graph[level])
endfor
TreeInitialization(initial pheromone intensity)

For level = L downto 1 do
                                                 Phase 3
     While not current level ending condition do
        For all ants do
           FindPath(probability rule)
        endfor
        UpdatePheromone(all ants paths vertices)
        DeamonAction(best path)
        EvaporatePheromone(all vertices)
     endwhile

                                                 Phase 4
     Refinement(graph[level])
endfor

                                                    Phase 5
LocalOptimization(best solution)
```

3. Electro-motor Power Losses Problem

Most home appliances with small electro-motors, such as vacuum cleaners or mixers, are driven by a universal motor. Because of the widespread use of the universal motor it is very important that the energy consumption of the motor (input power) is as low as possible, while still satisfying the needs of the user (output power). To ensure as high efficiency as possible the power losses of the motor should be as low as possible. In our case the power losses are reduced by optimizing the geometry of the stator and the rotor laminations. Because of the high magnetic saturation of the iron in universal motors, this is a highly non-linear problem.

The geometry of the rotor and the stator was defined parametrically (40 parameters). Some of the parameters were invariable, and were not altered for technological reasons. Only the dimensions of the variable parameters that were mutually independent were varied (10 parameters). In addition to the abovementioned independent variables there are also dependent variables, which were defined from independent ones. Because of many possible combinations of parameter values the problem is also very complex.

So, to find an optimal solution (the geometry of the motor) we need to find the values of the ten independent parameters $P_1, P_2, \ldots, P_{10}$ that define the rotor and the stator laminations (the thickness of the horizontal/vertical of the stator's yoke, the width of the stator, the radius of the stator's internal edge, the thickness of the stator's yoke at the hole, the length between the bisector and the slot edge, the radius of the stator's teeth, the external radius of the rotor, the width of the rotor's pole, and the height of the rotor's teeth).

4. Experimental Results

The MASA was run 20 times. On each run the algorithm coarsened the graph to level $L = 7$, and on each level of optimization we let 200 ants down the graph.

Table 1: Evaluation results.

	Power losses [W]				avg additional evaluations
	min	max	avg	±σ	
MASA w/o LO	114.23	135.90	128.86	7.78	N/A
MASA	110.40	134.17	126.21	9.58	116.25

So, overall the algorithm made 1400 evaluations (calculated with the ANSYS finite-element program) on each run. At the end, local optimization (LO) was applied, which required, on average, an additional

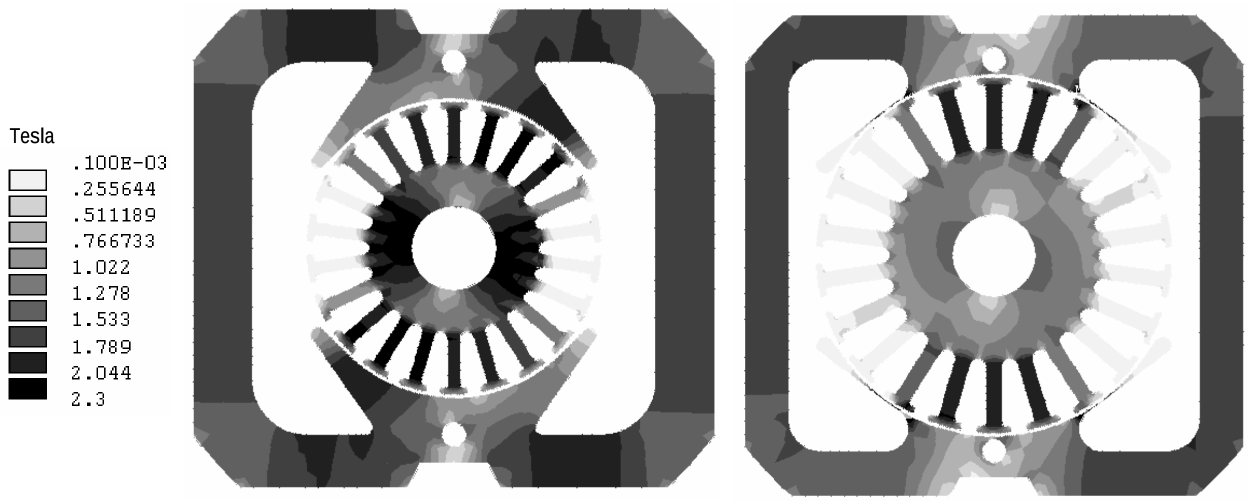

Fig. 4: Stator/rotor lamination designed by expert engineers (left), stator/rotor lamination designed by MASA (right).

116.25 evaluations. The time required for a run was approximately one day on an Athlon XP 1800+ processor. Almost all the time is consumed by the solution evaluations. The results are described in Table 1.

In addition, we compared the best solution obtained with the MASA to solutions found with expert engineers and genetic algorithm [6]. As shown in Table 2, we improved on the expert's solution by 44.3%, and on the GA's solution by 24.9%.

Table 2: Comparison of solutions.

	Expert	GA	MASA w/o LO	MASA
best solution [W]	198.35	147.00	114.23	110.40
improvement [%]	—	25.9	42.4	44.3

Figure 4 shows a stator/rotor lamination designed by expert engineers and by MASA. A comparison of the magnetic flux densities in the expert- and the MASA-designed motor shows a clear reduction in the areas with the highest levels of magnetic flux density in the MASA-designed motor.

5. Conclusion

In this paper we presented a new approach to solving multi-parameter problems based on stigmergy, a type of collective work that can be observed in an ant colony.

We proposed a general approach for the translation of a multi-parameter problem into a graph representation. Each path on the graph represents one possible solution, and the whole graph represents the whole solution space of the multi-parameter problem. For an efficient search of the solution space we used a multilevel paradigm. We call this approach the Multilevel Ant Stigmergy Algorithm.

The algorithm was tested on a real industrial problem: the minimization of the power losses in a universal electro-motor. The average solution obtained with our algorithm was better than a solution recently found using a genetic algorithm.

6. References

[1] P.-P. Grassé, "La reconstruction du nid et les coordinations inter-individuelle chez *Bellicositermes Natalensis et Cubitermes sp*. La théorie de la stigmergie: essai d'interprétation des termites constructeurs," *Insectes Sociaux*, vol. 6, pp. 41-83, 1959.

[2] M. Dorigo, G. Di Caro, and L. M. Gambardella, "Ant algorithms for discrete optimization," *Artificial Life*, vol. 5, pp. 137-172, 1999.

[3] G. Karypis, V. Kumar, "Analysis of multilevel graph partitioning," *Proc. of the ACM/IEEE Supercomputing Conference*, vol. 1, pp. 658-677, 1995.

[4] P. Korošec, J. Šilc, and B. Robič, "Solving the mesh-partitioning problem with an ant-colony algorithm," *Parallel Computing*, vol. 30, pp. 785-801, 2004.

[5] C. Walshaw, "Multilevel refinement for combinatorial optimisation problems," *Annals of Operations Research*, vol. 131, pp. 325-372, 2004.

[6] G. Papa, B. Koroušić-Seljak, B. Benedičič, and T. Kmecl, "Universal Motor Efficiency Improvement using Evolutionary Optimization," *IEEE Transactions on Industrial Electronics,* vol. 50, pp. 602-611, 2003.