Ant-Based Approaches for Solving Autocorrelation Problems

Ilias S. Kotsireas[1], Konstantinos E. Parsopoulos[2],
Grigoris S. Piperagkas[2], and Michael N. Vrahatis[3]

[1] Department of Physics and Computer Science, Wilfrid Laurier University,
Waterloo, Canada
ikotsire@wlu.ca

[2] Department of Computer Science, University of Ioannina, Greece
{kostasp,gpiperag}@cs.uoi.gr

[3] Department of Mathematics, University of Patras, Greece
vrahatis@math.upatras.gr

Abstract. We propose two ant–based formulations for solving autocorrelation problems. The formulations are combined with different ACO variants. Preliminary experiments of the derived approaches are conducted on two hard instances of the problem. Their performance is compared to an efficient Tabu Search algorithm, offering useful conclusions and motivation for further investigation.

1 Introduction

Several difficult combinatorial problems can be defined in a succinct way via the concepts of *periodic* and *non–periodic autocorrelation functions* (PAF and NPAF, respectively) associated with a finite binary or ternary sequence. Various metaheuristics have been previously employed in the search for solutions of similar combinatorial problems, with varying degrees of success.

The present paper aims at triggering the interest of the Ant Colony Optimization (ACO) research community in solving such combinatorial problems through ant–based approaches. For this purpose, we propose two ant–based formulations. To the best of our knowledge, this is the first attempt of formulating such problems in the specific algorithmic framework. As case study, we use our formulations to find *Hadamard matrices with two circulant cores*, a problem that can be defined via the PAF associated to two binary sequences. Nevertheless, the proposed ant–based formulations can be applied with minor modifications to any combinatorial problem defined via PAF and NPAF.

The sequences that arise as solutions to these problems are useful in a wide variety of applications, ranging from code–division multiple–access (CDMA) communication systems to pulse compression of radar signals. The reader is referred to [6] and [8] for further details and application areas. Additional applications in Coding Theory can be found in [3]. The rest of the paper is organized as follows: Section 2 offers brief descriptions of the problems while Section 3 introduces our ant–based formulations. Experimental results are presented in Section 4. The paper concludes with Section 5.

M. Dorigo et al. (Eds.): ANTS 2012, LNCS 7461, pp. 220–227, 2012.

2 Autocorrelation Problems

Let n be an odd positive integer. A finite sequence of length n with elements from the alphabet $\{-1, +1\}$ is called a *binary sequence.* Similarly, a finite sequence of length n with elements from the alphabet $\{-1, 0, +1\}$ is called a *ternary sequence.* The PAF associated to a finite sequence $\mathbf{a} = (a_1, \ldots, a_n)$ of length n, is a sequence $(\mathit{PAF}(\mathbf{a}, 0), \mathit{PAF}(\mathbf{a}, 1), \ldots, \mathit{PAF}(\mathbf{a}, n-1))$, also of length n where,

$$\mathit{PAF}(\mathbf{a}, s) = \sum_{i=1}^{n} a_i a_{i+s}, \qquad s = 0, 1, \ldots, n-1. \tag{1}$$

The quantity $i + s$ is taken modulo n whenever $i + s > n$. On the other hand, the NPAF associated to the finite sequence $\mathbf{a}$ is defined as $(\mathit{NPAF}(\mathbf{a}, 0), \mathit{NPAF}(\mathbf{a}, 1), \ldots, \mathit{NPAF}(\mathbf{a}, n-1))$ where,

$$\mathit{NPAF}(\mathbf{a}, s) = \sum_{i=1}^{n-s} a_i a_{i+s}, \qquad s = 0, 1, \ldots, n-1. \tag{2}$$

Chapter 7 of [3] contains a comprehensive description of the properties of the two autocorrelation functions defined above.

Given two finite (binary or ternary) sequences $\mathbf{a}$ and $\mathbf{b}$ of length n, we can request that their respective *PAF* or *NPAF* values (with the exception of the value at $s = 0$) add up to a constant, i.e.:

$$\mathit{PAF}(\mathbf{a}, s) + \mathit{PAF}(\mathbf{b}, s) = c_{\mathrm{PAF}}, \qquad \mathit{NPAF}(\mathbf{a}, s) + \mathit{NPAF}(\mathbf{b}, s) = c_{\mathrm{NPAF}}, \tag{3}$$

where $s = 1, 2, \ldots, n-1$. In general, sequences that satisfy these requirements are called *complementary.* When the values of the constants c_{PAF} and c_{NPAF} are small, i.e., less than 2 in absolute value, such sequences are said to exhibit *low (auto)correlation* and they have important engineering applications [6].

There are some particularly important cases of combinatorial problems that can be defined via the aforementioned setup. In our work, we focused on the case of binary sequences with $c_{\mathrm{PAF}} = -2$, which are related to *Hadamard matrices with two circulant cores* or equivalently to *Generalized Legendre pairs* [5]. The solution sequences of the aforementioned problems are also subject to *Diophantine Equations* (DEs) that shall be satisfied by the solutions. For instance, in our case study the corresponding DE is:

$$sa^2 + sb^2 = 1, \tag{4}$$

where $sa = a_1 + \cdots + a_n$ and $sb = b_1 + \cdots + b_n$. Equation (4) has four solutions, up to sign. In our study we focus on the case where $sa = 1$ and $sb = 1$.

3 Proposed Ant–Based Approaches

In the following paragraphs, we introduce two different formulations of autocorrelation problems within the framework of ant–based algorithms.

3.1 Approach Based on Binary Ants

Let n be the length (positive odd integer) of each one of the two sequences and let $N = 2n$. Then, a candidate solution has the following form:

$$\mathbf{x} = \Bigg(\underbrace{x_1, x_2, \ldots, x_n,}_{\text{sequence } \mathbf{a}} \quad \underbrace{x_{n+1}, x_{n+2}, \ldots, x_N}_{\text{sequence } \mathbf{b}} \Bigg)^{\top}. \tag{5}$$

The corresponding optimization problem is N–dimensional and it is defined as the *minimization* of the objective function:

$$F(\mathbf{x}) = \left| \sum_{i=1}^{n} a_i - 1 \right| + \left| \sum_{i=1}^{n} b_i - 1 \right| + \sum_{s=1}^{(n-1)/2} \Big| \mathit{PAF}(\mathbf{a}, s) + \mathit{PAF}(\mathbf{b}, s) + 2 \Big|, \tag{6}$$

where $\mathit{PAF}(\mathbf{a}, s)$ and $\mathit{PAF}(\mathbf{b}, s)$ are defined according to Eq. (1). In order to put the problem in the ACO framework we shall define a table of pheromones, i.e., weights for the possible component values of a candidate solution. The probabilities of selecting -1 or $+1$ for the component x_i of a new candidate solution $\mathbf{x}$, are defined as follows:

$$p_{i,(-)} = \frac{w_{i,(-)}^{\alpha} \eta_{i,(-)}^{\beta}}{w_{i,(-)}^{\alpha} \eta_{i,(-)}^{\beta} + w_{i,(+)}^{\alpha} \eta_{i,(+)}^{\beta}}, \qquad p_{i,(+)} = \frac{w_{i,(+)}^{\alpha} \eta_{i,(+)}^{\beta}}{w_{i,(-)}^{\alpha} \eta_{i,(-)}^{\beta} + w_{i,(+)}^{\alpha} \eta_{i,(+)}^{\beta}}, \tag{7}$$

where $w_{i,(-)}$, $w_{i,(+)}$, are the corresponding pheromone levels (weights) for the values -1 and $+1$; α and β are user–defined parameters controlling the strength of each term; and η is a function of desirability of the corresponding component, i.e., it defines its significance in the solution vector.

The value of the component x_i is determined by drawing a uniformly distributed random number, i.e., $x_i = -1$, if $\texttt{rand}() < p_{i,(-)}$, otherwise $x_i = +1$. The same selection procedure is independently applied for all components. Instead of one, K ants can be used to construct K candidate solutions at each iteration. We will call this set the *colony* in our approach. Nevertheless, the construction procedure is identical for all ants.

If t denotes the iteration number, the colony can be denoted as $\mathbf{S}^{(t)} = \left\{ \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}, \ldots, \mathbf{x}_K^{(t)} \right\}$ with $\mathbf{x}_i^{(t)} = \left(x_{1,i}^{(t)}, x_{2,i}^{(t)}, \ldots, x_{N,i}^{(t)} \right)^{\top}$, $i = 1, 2, \ldots, K$. After generating the K candidate solutions of the t–th iteration, their evaluation with the objective function $F(\mathbf{x})$ of Eq. (6) takes place and the pheromones are updated as follows:

$$w_{i,j}^{(t+1)} = (1 - \rho) w_{i,j}^{(t)} + \mathit{\Delta} w_{i,j,k}^{(t)}, \qquad j \in \{-1, +1\},\ i = 1, 2, \ldots, K, \tag{8}$$

where $\mathit{\Delta} w_{i,j,k}^{(t)} = V_k^{(t)}$, if $x_{k,i}^{(t)} = j$; otherwise $\mathit{\Delta} w_{i,j,k}^{(t)} = 0$, $k = 1, 2, \ldots, N$. The increment $V_k^{(t)}$ can be either fixed for all ants or inversely proportional to the objective value of each contributing ant, i.e., $V_k^{(t)} = 1/F(\mathbf{x}_i^{(t)})$ or $V_k^{(t)} = Q$.

Different ant–based algorithms are distinguished by considering different groups of contributing ants. In *Ant Colony Optimization* [1] (ACO) only new ants produced at each iteration contribute to the pheromone update. In *Elitist Ant System* [4] (EAS), in addition to the currently constructed ants, also the overall best ant contributes to the pheromones. In the *Max–Min Ant System* [7] (MMAS) scheme the pheromones are bounded within a range $[w_{\min}, w_{\max}]$ and initialized to their maximum values. Also, they are re–initialized to their maximum value whenever stagnation is detected. In contrast to the previous variants, only the best ant of the current iteration or the overall best ant deposits pheromones (we considered only the latter case).

These three popular variants were also considered in our experiments with two minor modifications. Specifically, we adopted a fixed lower bound for the pheromones in all variants (not only for MMAS). This decision aimed at avoiding the actual elimination of component values with very small pheromones in the selection procedure. Also, the algorithm was restarted whenever there was no improvement of the overall best solution for a predefined number of function evaluations. This number was defined as a fraction of the maximum available computational budget. Apart from these modifications, some additional alterations proved to enhance the algorithm's performance and they are described in the following section.

3.2 Performance Enhancing

The construction of candidate solutions can be modified to ensure that the produced solutions will be feasible with respect to the DEs, i.e., the sums of their components will be equal to 1 for both their sequences. Specifically, each sequence has length n and constitutes of values in $\{-1, +1\}$. Hence, we can build a candidate solution by determining its components in pairs of complementary values, i.e., if one component receives -1 then another randomly selected component automatically receives $+1$. At the end, there will be one remaining component that is set directly to $+1$. Thus, -1 appears in $(n-1)/2$ components while $+1$ appears in $(n+1)/2$ components.

This procedure produces candidate solutions that, by construction, satisfy the DEs. Therefore, the first two terms terms in the objective function can be dropped, resulting in the following form:

$$F(\mathbf{x}) = \sum_{s=1}^{(n-1)/2} \Big| PAF(\mathbf{a}, s) + PAF(\mathbf{b}, s) + 2 \Big|, \tag{9}$$

that replaces the one defined in Eq. (6). The performance of the algorithm can be further enhanced by incorporating local search. We adopted the procedure used in [2] within the framework of a very efficient Tabu Search approach.

3.3 Approach Based on Components Permutation

An alternative ant–based approach, yet closer to the general principles of the original ACO algorithms, is based on the formulation of the problem as a search procedure for the best permutation of components. More specifically, we can define a candidate solution as a permutation of solution components indices:

$$\mathbf{x} = \left(\underbrace{c_1^{\mathbf{a}},\ c_2^{\mathbf{a}},\ \ldots\ ,\ c_n^{\mathbf{a}}}_{\text{permutation for seq. } \mathbf{a}}\ ,\quad \underbrace{c_1^{\mathbf{b}},\ c_2^{\mathbf{b}},\ \ldots\ ,\ c_n^{\mathbf{b}}}_{\text{permutation for seq. } \mathbf{b}} \right)^{\top}, \tag{10}$$

where $c_j^{\mathbf{a}} = i$, $i \in \{1, \ldots, n\}$, denotes that the i–th component of the actual (binary) sequence $\mathbf{a}$ possesses the j–th position in the specific permutation defined by $\mathbf{x}$. Then, we build a binary vector by translating $\mathbf{x}$, assuming that components that appear in the first $(n+1)/2$ positions of the permutation are assigned the value $+1$, while the rest are assigned the value -1, i.e.:

$$\mathbf{y_x} = \left(\underbrace{a_1, a_2, \ldots, a_n}_{\text{sequence } \mathbf{a}},\quad \underbrace{b_1, b_2, \ldots, b_n}_{\text{sequence } \mathbf{b}} \right)^{\top}, \quad a_i, b_i \in \{-1, +1\},\ i = 1, 2, \ldots, n. \tag{11}$$

where $a_i\,(\text{resp.}\, b_i) = +1$, if $c_j^{\mathbf{a}}\,(\text{resp.}\, c_j^{\mathbf{b}}) = i$ for j such that $1 \leqslant j \leqslant (n+1)/2$; otherwise $a_i\,(\text{resp.}\, b_i) = -1$. Apparently, this permutation–based representation of the ants requires also different pheromone table representation than that of Section 3.1. Indeed, a pheromone entry $w_{i,j}^{\mathbf{a}}$ for sequence $\mathbf{a}$ defines the weight (pheromone level) for the case where component index j appears immediately after index i in the permutation of sequence $\mathbf{a}$ of a candidate solution. The corresponding quantities are defined by weights $w_{i,j}^{\mathbf{b}}$ for sequence $\mathbf{b}$.

The objective value of the ant $\mathbf{x}$ is defined through its corresponding binary translation $\mathbf{y_x}$, i.e., $F(\mathbf{x}) = F(\mathbf{y_x})$. Since, by definition, the DEs hold for the produced translated vectors $\mathbf{y_x}$, we can use the objective function defined in Eq. (9) instead of Eq. (6). The same ant–based variants along with all modifications and performance enhancing techniques mentioned in the previous section, were also used with this formulation.

4 Experimental Results

We report indicative experimental results for the ant–based approaches defined in the previous sections for PAF problems of length $n = 29$ and 39. We shall mention that $n = 77$ is the smallest value for which this is an open problem. Henceforth, we will denote as "Bin" the approach based on binary ants and "Per" the one based on permutations. The basic ACO algorithm will be denoted as "A", the EAS as "B" and the MMAS as "C". Finally, the fixed pheromone increment approach will be denoted as "a", while the proportional one as "b".

Regarding the parameter setting, the maximum function evaluations was $\text{fev}_{\max} = 200 \times 10^6$, while the maximum evaluations for restart (if no improvement) was $R = \text{fev}_{\max}/5$. The pheromone scaling factor of Eq. (8) was $\rho = 0.1$ and the fixed pheromone increment was set to $Q = 0.001$. The pheromones were bounded in the range $[0.01, 1.0]$, while the parameters η, α and β in Eq. (7) were

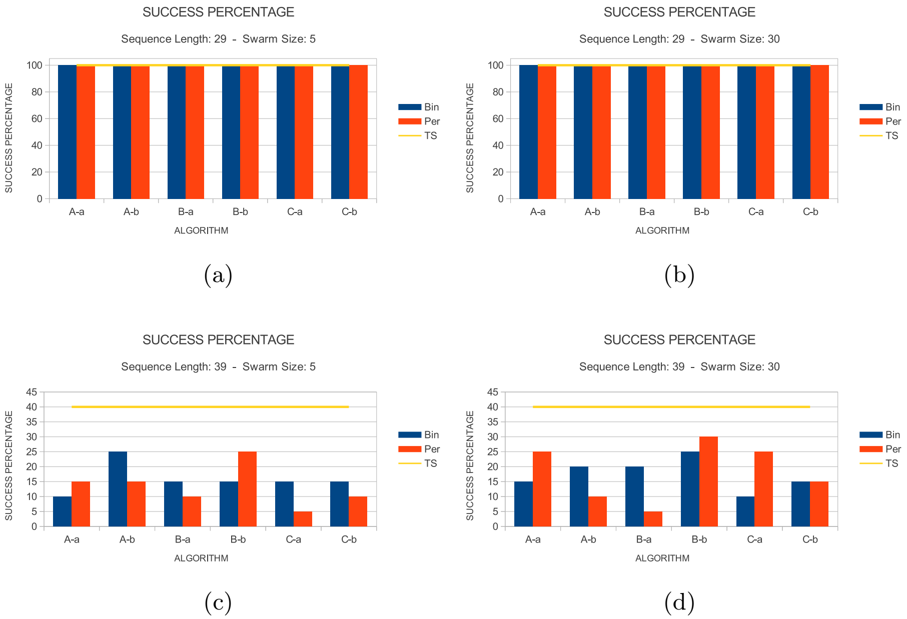
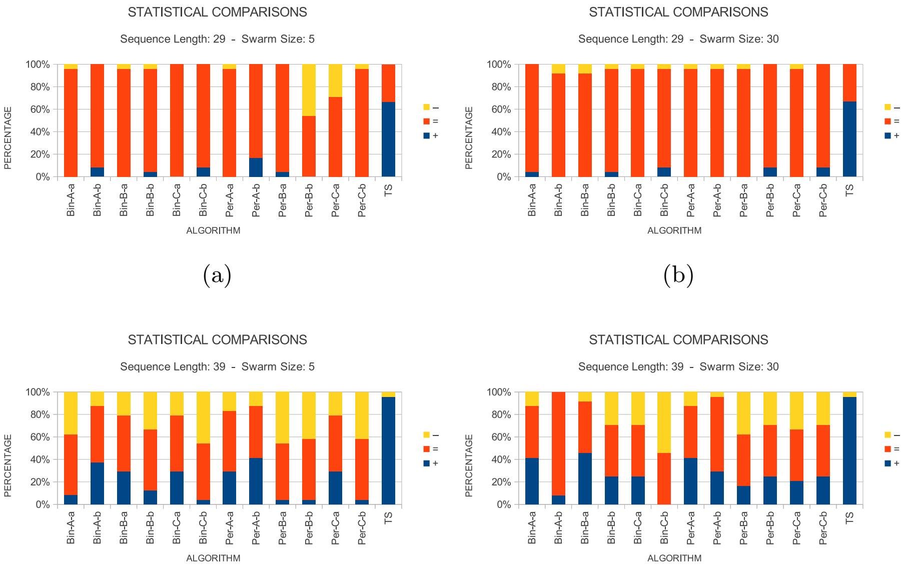


Fig. 1. Success percentages per problem and swarm size

all set to 1. Finally, two numbers of ants were considered, namely 5 and 30. The number of experiments conducted per approach was 20. Each experiment was terminated as soon as a global minimizer was found or the available computational budget was exceeded. An experiment was considered as successful only if an optimal solution was detected.

In order to provide a measure of performance for the proposed ant–based approaches, we performed the same number of independent experiments also for the TS approach proposed in [2] for the same computational budget. For comparison purposes, we also conducted Wilcoxon rank–sum tests for each pair of algorithms per problem and swarm size, and we recorded the number of favorable (denoted with "+"), unfavorable (denoted with "-") and neutral (denoted with "=") comparisons for 95% significance level. However, we shall underline that no effort was paid in fine–tuning the ant–based approaches in the comparisons, because a thorough comparison among the different methodologies was out of the scope of the present paper. Thus, we used the TS performance mostly as a reference point for a preliminary assessment of the proposed methods.

The results are graphically illustrated in Figs. 1 and 2. Specifically, Fig. 1 illustrates the success percentage of each algorithm per problem and swarm size, i.e., the percentage of experiments where it managed to detect a global optimizer within the available computational budget. The bars refer to the Bin and Per ant–based approaches, while the corresponding TS performance for the same experimental setting is depicted as a horizontal line. On the other hand, each bar



(a) (b)

(c) (d)

Fig. 2. Statistical comparison tests among the algorithms

in Fig. 2 illustrates the percentage of positive, negative and equal comparisons of the corresponding approach with the rest of the algorithms.

A close inspection of the results offers intuition on the algorithms' performance. Regarding the success percentages of the algorithms, as we see in Fig. 1, all approaches were completely successful for the case of sequence length $n = 29$. However, the picture radically changes for the (much harder) problem of $n = 39$. The performance of all algorithms, including TS, plunges by at least 40%. Moreover, the ant–based approaches exhibit also different behavior with respect to the swarm size. For the case of $K = 5$ ants, the Bin approaches have superior performance than Per for the 4 out of 6 variants, as depicted in Fig. 1(c). Yet, the Per approaches outperformed Bin for the A–a (ACO with fixed pheromone increments) and B–b (EAS with proportional increments) cases. Increasing the swarm size to $K = 30$, significantly improves the performance of Per approaches, especially for the C variant (MMAS). Also, Per remains better than Bin for A–a and B–b. However, even the Bin approaches achieve better performance in 3 out of 6 variants, compared to the 5 ants case. Thus, the first impression is that higher swarm size can rise the probability of successful experiments.

Inspecting Fig. 2, we can verify that the case $n = 29$ can be efficiently solved by all algorithms. In most cases, the algorithms exhibit statistically insignificant differences in performance but outperformed by TS in almost 60% of the cases. Again, increasing swarm size seems to produce essentially identical performance among ant–based approaches. However, the picture becomes more complicated in the $n = 39$ case. In this cases, the approaches with fixed pheromone increments have a clear increasing trend of positive comparisons for ACO and EAS. On the other hand, some of the approaches with proportional increments seem to loose part of their efficiency, especially for Bin. This behavior can be ascribed to the faster biasing towards the best performing ants offered due to the proportional increments, in combination with the higher dimensionality and degree of difficulty.

5 Conclusions

We presented two different ant–based formulations for tackling autocorrelation problems. Various combinations with different ACO–based approaches were considered and tested on two problems of different dimensionality and degree of difficulty. The proposed approaches were compared against a specialized TS approach with verified efficiency. The results are promising, offering space for further improvement by proper fine–tuning of the ant–based approaches. Also, they reveal that swarm size can play a role in the algorithms' performance profiles. However, the exact tendency of each approach remains to be investigated in depth. This will be the main subject of our future work.

References

1. Bonabeau, E., Dorigo, M., Théraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, New York (1999)
2. Chiarandini, M., Kotsireas, I.S., Koukouvinos, C., Paquete, L.: Heuristic algorithms for Hadamard matrices with two circulant cores. Th. Com. Sc. 407, 274–277 (2008)
3. Colbourn, C.J., Dinitz, J.H. (eds.): Handbook of combinatorial designs. Discrete Mathematics and its Applications. Chapman & Hall/CRC, Boca Raton (2007)
4. Dorigo, M., Maniezzo, V., Colorni, A.: Ant system: Optimization by a colony of cooperating agents. IEEE Trans. SMC – Part B 26(1), 29–41 (1996)
5. Fletcher, R.J., Gysin, M., Seberry, J.: Application of the discrete Fourier transform to the search for generalised Legendre pairs and Hadamard matrices. Australas. J. Combin. 23, 75–86 (2001)
6. Golomb, S.W., Gong, G.: Signal design for good correlation. Cambridge University Press, Cambridge (2005)
7. Stützle, T., Hoos, H.H.: Max–min ant system. Future Generation Computer Systems 16, 889–914 (2000)
8. Tran, L.C., Wysocki, T.A., Mertins, A., Seberry, J.: Complex Orthogonal Space-Time Processing in Wireless Communications. Springer (2006)