COMPARISON OF ANT COLONY OPTIMISATION AND DIFFERENTIAL EVOLUTION

Karl O. Jones, André Bouffet

***Abstract**:* Evolutionary Computation (EC) is a fairly new and vibrant area of investigation. Some of the least widely known approaches within EC are Ant Colony Optimisation and Differential Evolution, both of which can be used in optimisation problems. Ant Colony Optimisation is based on the foraging patterns of ants, while Differential Evolution is based on an Evolutionary Algorithm that uses a 'greedy' selection scheme. The problem area chosen is that of identification of model parameters (as used in control engineering).

***Keywords**:* Ant colony optimisation, differential evolution, process modelling, simulation*.*

INTRODUCTION

Evolutionary Computation (EC) algorithms are part of the artificial intelligence field, and often take their principles from natural mechanisms, such as animal behaviour. The most famous of these evolutionary algorithms is the Genetic Algorithm (GA) based on Darwin's Theory of Evolution. The GA reproduces evolution laws on a population that represents the solutions to a specific problem. GAs have been applied to numerous problems, and are today known as an efficient optimisation method which is commonly used in industry. GAs have been applied to PID tuning and have produced very effective results [1]. Other schemes use the cooperation between agents, similar to social animal behaviour thus if a single agent is not able to perform a task, an associated individual can. One such approach is Ant Colony Optimisation (ACO) which is founded on the foraging behaviour of ants and their indirect communication based on pheromones. ACO has been applied to several combinatorial problems such as job scheduling [2], routing optimization in data communication networks [3] and telephone networks [4].

Differential Evolution (DE) was first introduced by Price and Storn in 1995 [5][6]. DE can be classified as an evolutionary optimization algorithm, and was developed from Price's attempts to solve the Chebychev Polynomial fitting Problem. DE is an exceptionally simple evolution strategy that is significantly faster and robust at numerical optimisation.

There are many control systems that require some form of mathematical model of the process, for example model-based predictive control. Occasionally accurate models can be analytically derived through consideration of known physical processes. This approach is often suitable for linear, deterministic, time-invariant, single-input single-output systems, where knowledge of the system processes is available. However, most real-world systems do not fit into this category; in particular they are often non-linear and poorly understood. Hence "system identification", or Black-box modelling, is often the only realistic approach available. System modelling can be decomposed into two inter-related, problems: *Selection* of a suitable model structure, and *Estimation* of the model parameters. The work presented here focuses on *Estimation*.

ANT COLONY OPTIMISATION (ACO)

ACO is based on ant ability to always find the shortest path between their nest and a food source. Several experiments have been undertaken to study this behaviour, where different length paths were constructed between a nest and a food source, and the number of ants following each path monitored. Initially paths are chosen at random, however over time more and more ants follow the shorter path (Figure 1: Initial position (a) and final situation (b)).

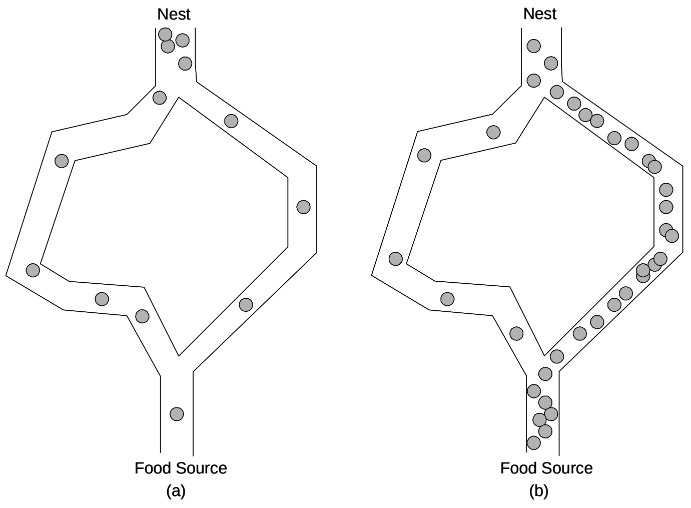

Figure 1. Ants Following Pheromone Trail.

An ACO is built in three phases (see Table 1): firstly, the specific problem is mapped in graph form, secondly a tour of solution is created, lastly the pheromone is updated.

Table 1. ACO operation (modified from [7])

```
procedure ACOstatic
                create graph solution,
                set parameters
                initialise pheromones
    while (termination condition not met) do
                Construct ant solution (Tour)
                Update pheromones
    end
end
```

The problem must be mapped into a weighted graph, so the ants can cover the problem to find a solution. The ants are driven by a probability rule to choose their solution to the problem, known as a tour. The probability rule (equation 1) between two nodes i and j, called Pseudo-Random-Proportional Action Choice Rule, depends on two factors: the heuristic and metaheuristic

$$p_{ij} = \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{h \in s}[\tau_{ih}]^{\alpha}[\eta_{ih}]^{\beta}} \quad (1)$$

The heuristic factor η_{ij}, or visibility, is related to the specific problem as the inverse of the cost function. This factor does not change during algorithm execution, instead the metaheuristic factor τ_{ij} (related to pheromone) is updated after each iteration. The pheromone must have an initial value τ_0. The parameters α and β enable the user to direct the algorithm search in favour of the heuristic or the pheromone factor. These two factors are dedicated to every edge between two nodes and weight the solution graph.

The pheromones are updated after a tour is built, in two ways: firstly, the pheromones are subject to an evaporation factor (ρ), which allows the ants to forget their past and avoid being trapped in a local minima (equation 2). Secondly, they are updated in relation to the quality of their tour (equations 3 and 4), where the quality is linked to the cost function.

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij}, \qquad \forall (i,j) \in L \tag{2}$$

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}, \qquad \forall (i,j) \in L \tag{3}$$

$$\Delta\tau_{ij}^{k} = \begin{cases} 1/c^{k} & \text{if arc (i,j) belongs to } T^{k} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where m is the number of ants, L represents the edges of the solution graph, and C^k is the cost function of tour T^k, built by the kth ant.

DIFFERENTIAL EVOLUTION (DE)

DE is a very simple population based, stochastic function minimizer which at the same time is very powerful

Step 1: DE starts with a population of NP randomly chosen solution vectors. For each population member, *i*, a 'mutant vector' is formed:

$$v_i = x_{r1} + F \cdot (x_{r2} - x_{r3}) \tag{5}$$

Where r_1, r_2 and r_3 are three mutually distinct randomly drawn indices from the population NP, and also distinct from *i*. The real and constant factor F, controls the amplification of the differential variation, where $0<F\leq 2$.

Step 2: A process of crossover is performed. For each component of the vector, draw a random number in U[0,1] which is called $rand_j$. Let $0\leq CR<1$ be a cut-off. If $rand_j \leq CR$ then $u_{ij}=v_{ij}$ else $u_{ij}=x_{ij}$. To ensure that at least some form of crossover takes place, one component of u_i is selected at random to be from v_i.

Step 3: If the objective value COST(u_i) is lower than COST(x_i), then u_i replaces x_i in the next generation. Otherwise, we keep x_i.

Step 4: Repeat Steps 2 and 3 until the size of the new population equals that of initial population, NP.

Step 5: Replace the initial (parent) population with the new (offspring) population.

Step 6: Iterate the process until the termination criterion is satisfied.

SYSTEM MODELLING RESULTS

Simulations were conducted on a Celeron 1.7GHz computer, using the MATLAB 7 environment. Two processes were used for the identification task based on the transfer function:

$$G(s) = \frac{Ke^{-T_d s}}{T_2 s^2 + T_1 s + 1} \tag{6}$$

The first process selected was given by:

$$G(s) = \frac{7.6e^{-25s}}{10.3s^2 + 1s + 1} \tag{7}$$

where the optimisation algorithms had to identify K, T_d and T_2 only. The second system was represented by:

$$G(s) = \frac{5.7e^{-39s}}{40.2s^2 + 6.2s + 1} \tag{8}$$

where the algorithms had to identify K, T_d, T_1 and T_2. The known systems were simulated to create a training data set which could be compared with the model output data set produced by the parameters proposed by the optimisation algorithms. The comparison was conducted using the Integral Absolute Error (IAE) as the objective function to be minimised. Since real system measurements are rarely smooth, on some tests a perturbation was added to the system response to represent noise (25% of the process signal). For both algorithms the population was set to 25 individuals, with a maximum of 100 generations. The results of applying the DE and ACO to the identification problem are provided in Table 1. For each parameter the final value determined by the respective algorithm is given followed by its percentage difference from the known value: for example for the first system, without noise, the ACO determined T_1 as 6.60, 13.1% above the actual value of 10.3. The second to last column presents the number of generations taken to arrive at the determined parameter value, while the final column reports the number of seconds required by the CPU for the simulation (at most 100 generations): the time was determined using the MATLAB function *cputime*. In all cases the ACO computation effort significantly exceeds that of the DE.

Table 2 Parameter Identification Variation

	K		**T_1**		**T_2**		**T_d**		**No. of Gen.**	**CPU Time** (sec)
	Value	**% diff.**	**value**	**% diff.**	**value**	**% diff.**	**value**	**% diff.**		
Ant Colony Optimisation										
No noise	6.600	13.1%	10.50	-1.94%	*n/a*	*n/a*	25.60	-2.40%	100	556.1
With noise	7.900	-3.94%	7.60	26.2%	*n/a*	*n/a*	27.50	-10.0%	100	500.6
No noise	5.200	8.77%	44.90	-11.7%	10.00	-61.3%	31.00	20.5%	100	760.2
With noise	5.8	-1.75%	43.60	-8.45%	37.00	496.7%	39.20	-0.51%	100	763.3
Differential Evolution										
No noise	7.600	0%	10.30	0%	*n/a*	*n/a*	25	0%	75	135.4
With noise	7.602	-0.02%	10.29	0.02%	*n/a*	*n/a*	25.01	-0.03%	80	206.1
No noise	5.699	0.01%	41.52	-3.30%	6.35	-2.46%	38.73	0.67%	100	203.6
With noise	5.705	-0.09%	40.75	-1.37%	6.24	-0.70%	38.96	0.08%	100	196.3

The typical change in Objective Function value over the generations can be seen in Figure 2, illustrates the searching and optimisation processes undertaken by both algorithms. It will be noticed that the fitness function value for the DE algorithm follows a steady downward path, while that for the ACO has a more erratic trajectory. This is a classic result from the ACO since its evolution is based on the pheromone levels left by the ants and hence does not contain such a direct link to knowledge contained in previous iterations as methods such as genetic algorithms and DE.

During the work, it was noted that the ACO did not always locate the global optimum. In fact, the desired values were only achieved approximately half the time, with the remaining occasions producing values positioned close to the global optimum: called the pseudo global solution. To overcome this problem and to get satisfactory results, the ACO was run five times.

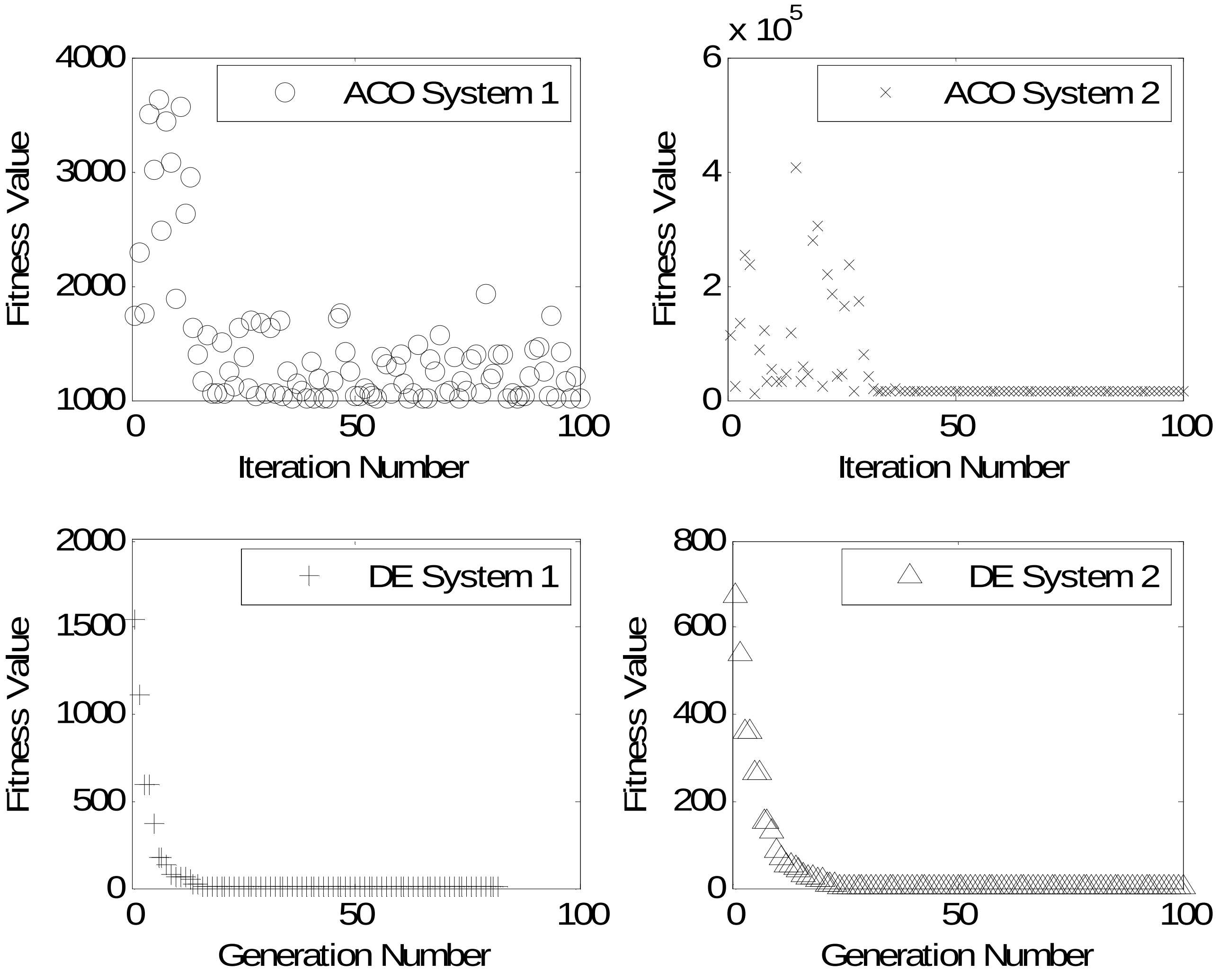

Figure 2 Fitness function over generations.

CONCLUSIONS

The strength of both ACO and DE is in the parallel nature of their search. We observed that ACO never converged to the global minimum, at best it produced 'ball park' answers. The ACO produced the global minimum around fifty percent of the time and for the balance, it is caught in, what we have called, the pseudo global minimum solution (PGS). In the PGS the solution obtained is located around the global minimum, where the gradient of the curve is very small. It produces a small difference between the fitness values that leads the ACO toward the wrong solution. This is a significant drawback of the ACO technique. One problem remains for both approaches, namely the appropriate selection of an objective function: although this is an issue with all optimisation techniques.

Overall the results indicate that both ACO and DE can be used for optimising model parameters during system identification. In computation terms, the DE approach is faster, although neither algorithm takes what can be considered an unacceptably long time to determine the results. In terms of the accuracy of the model parameters, the DE algorithm determines values which are exceptionally close to the known values. Additionally, the DE technique arrives at its final parameter values in fewer generations than the maximum permitted during the work. Thus it must be concluded that for the procedure of process modelling, the DE approach is superior to the ACO approach.

Any meta-heuristic should not be thought of in isolation: the possibility of utilising hybrid approaches should be considered. For example, the ACO approach might be more suited to determining the approximate region of the values from which a DE can be used to identify the precise value.

Techniques such as DE have shown themselves to be effective solutions to optimisation problems (ACO to a lesser extent). However, techniques such as ACO, DE and others are not a universal remedy, despite their apparent robustness. Each approach has control parameters set by the user, and appropriate setting of these parameters is a key point for success.

REFERENCES

[1] Herrero, J.M., *et al.* 2002. Optimal PID tuning with Genetic Algorithms for Non-Linear process Models. IFAC 15th Triennial World Congress, Barcelona, Spain.
[2] Colorni, A., M. Dorgio, V. Maniezzo and M. Trubian. 1994. Ant System for Job Scheduling. Belgian Journal of Operations Research, Statistics and Computer Science, (Vol. 34), p. 36-53.
[3] Di Car, G. and M. Dorgio. 1998. AntNet: Distributed Stigmergetic Control for Communication Networks. Joural of Artificial Intelligence Research, (Vol. 9), p. 317-365.
[4] Schoonderwoerd, R., O. Holland, J. Bruten and L. Rothkrantz. 1996. Ant-Based Load Balancing in Telecommunications Networks. Adaptive Behaviour, (Vol. 5), p. 169-207.
[5] Storn, R. and K. Price. 1995. Differential Evolution – A Simple and Efficient Adaptive Scheme for Global Optimisation over Continuous Spaces. Technical Report TR-95-012, ICSI. Available via ftp://ftp.icsi.berkeley.edu/pub/techreports/1995/tr-95012.ps.z
[6] Storn R. and K. Price. 1997. Differential Evolution – A Simple and Efficient Heuristic for Global Optimisation over Continuous Spaces. Journal of Global Optimisation, 11(4), 341-359.
[7] Dorigo, M. and Stützle, T. 2004. Ant Colony Optimization. The MIT Press, USA.

ABOUT THE AUTHORS

Dr. Karl O. Jones is a Principal Lecturer in the School of Engineering at Liverpool John Moores University, while Andre Bouffet was an MSc student working under his supervision. André now works in France for the Yokagawa organization.
Phone: +44 151 231 2199, E-mail: k.o.jones@ljmu.ac.uk.