Bayesian optimisation to select Rössler system parameters used in Chaotic Ant Colony Optimisation for Coverage

Martin Rosalie, Emmanuel Kieffer, Matthias R Brust, Gregoire Danoy, Pascal Bouvry

Bayesian optimisation to select Rössler system parameters used in Chaotic Ant Colony optimisation for Coverage

Martin Rosalie[a,b,c], Emmanuel Kieffer[a], Matthias R. Brust[a], Grégoire Danoy[a,*], Pascal Bouvry[a]

[a] *SnT, University of Luxembourg, Luxembourg*
[b] *Univ. Perpignan Via Domitia, Laboratoire Génome et Développement des Plantes, UMR5096, F-66860, Perpignan, France*
[c] *CNRS, Laboratoire Génome et Développement des Plantes, UMR5096, F-66860, Perpignan, France*

Abstract

The CACOC (Chaotic Ant Colony optimisation for Coverage) algorithm has been developed to manage the mobility of a swarm of Unmanned Aerial Vehicles (UAVs). Using a specific chaotic dynamic obtained from the Rössler system, CACOC provides waypoints for UAVs that aim to optimise the coverage of an unknown area while having unpredictable trajectories. Since the chaotic dynamics are obtained from a three differential equations system with parameters, it is possible to tune one parameter to obtain another chaotic dynamic, which will result in different UAV mobility behaviours. This work aims at optimising this parameter of the Rössler chaotic system to improve the coverage performance of CACOC. Since each evaluation of a solution requires a full simulation, global optimisation techniques (e.g., population-based heuristics) would be very time-consuming. We therefore considered a surrogate-based method to efficiently explore the parameter space of the Rössler system for CACOC, i.e., Bayesian optimisation. Experimental results demonstrate that this approach permits to improve the speed of coverage of the UAV swarm. In addition an analysis of the dynamical properties of the obtained chaotic system is provided.

Keywords: Bayesian optimisation, UAV swarm, Chaotic system

*Corresponding author

Email addresses: martin.rosalie@univ-perp.fr (Martin Rosalie), emmanuel.kieffer@uni.lu (Emmanuel Kieffer), matthias.brust@uni.lu (Matthias R. Brust), gregoire.danoy@uni.lu (Grégoire Danoy), pascal.bouvry@uni.lu (Pascal Bouvry)

1. Introduction

Several works have recently demonstrated that replacing random processes by chaotic dynamics can enhance the performance of optimisation algorithms [1, 2]. This approach was recently used in the field of Unmanned Aerial Vehicles (UAVs) with the CACOC (Chaotic Ant Colony optimisation for Coverage) [3]. This algorithm uses a specific chaotic dynamic obtained from the Rössler system [4] in order to optimise the coverage of an unknown area by a swarm of UAVs. The Rössler system has three parameters (a, b and c) leading to either periodic or chaotic solutions. A modification of these parameters can deeply impact CACOC results. However each evaluation of a solution requires a full UAV swarm simulation, which is computationally expensive. Global optimisation techniques such as population-based heuristics would require too many evaluations and are thus not applicable.

For that reason hyperparameter optimisation [5] has been considered, i.e., the optimisation of the parameters of complex systems or models. Contrary to standard optimisation approaches, they attempt to minimise an objective function while taking into account the evaluation time. Indeed, many theoretical models can be very time consuming to evaluate when changing their parameters. Therefore, most of the hyperparameter optimisation approaches do not only optimise an objective function but also try to minimise the number of steps required to reach a good solution.

Bayesian optimisation is one such methods, based on Gaussian processes (GP) and an acquisition function. The GP regressive model aims at approximating the unknown function by creating some surrogate function [6]. This function is iteratively refined by optimising the acquisition function until we are unlikely to find a new improving solution. The acquisition function is an auxiliary function taking into account statistical information about the presence of promising area. This function is computed using the current posterior distribution.

This work thus considers the usage of a surrogate-based method, i.e. bayesian optimisation, to efficiently explore the parameter space of the Rössler system for CACOC, in order to improve the resulting area coverage performance of the swarm.

In the remainder of this article, we first introduce the related work on the usage of chaotic dynamics in metaheuristics and of metaheuristics to select the best parameters of dynamical systems. Section 3 provides information on the optimisation problem, namely the CACOC mobility model, its Rössler system parameters and the evaluation function used. The bayesian optimisation approach is presented in Section 4. Section 5 provides the experimental setup and results analysis while Section 6 introduces a dynamical analysis to explain the numerical results. Finally, Section 7 proposes our conclusions and perspectives.

2. Related work

This section presents an analysis of the related work on the combination of metaheuristics and chaotic dynamics. The first subsection focuses on the usage of chaotic dynamics within metaheuristics as a way to replace random components in these algorithms, as used in CACOC. The second subsection provides a survey on the usage of metaheuristics to optimise chaotic systems, as proposed in this work.

2.1. Chaotic dynamics used in metaheuristics

To solve continuous or discrete global optimisation problems with chaotic dynamics, Tatsumi *et al.* [7] distinguish two types of methods. Most of the litterature focuses on the usage of well known chaotic functions such as the logistic map to replace random values in algorithms. The second type of method are chaotic metaheuristics based on a gradient method. For the latter, Toduka *et al.* [8] propose an algorithm to solve nonlinear optimisation problems by the construction of dynamical system. They present a global bifurcation scenario to realize a "chaotic search" where the bifurcation parameter is updated during the simulation to obtain a dynamical system that is able to minimise the objective function. Recently, Tatsumi *et al.* [9] detail that chaotic metaheuristics with the gradient method with perturbation have good performances for solving some benchmark problems and it can be enhanced via quasi-Newton method. The latter method permits to efficiently select parameters from their dynamical system in bifurcation diagrams to address classical optimisation problems.

As stated previously, most of these metaheuristics use chaotic dynamics to replace random numbers in evolutionary algorithms. However, the use of a given dynamical system in a metaheuristic cannot be efficient for all problems as stated in the "no free lunch" theorem [10]. Gandomi *et al.* [1] thus proposed to use several chaotic dynamics from discrete maps to accelerate a PSO (Particle Swarm optimisation) algorithm. The attraction parameter of the acceleration method is updated at each step using the value of various discrete maps. They numerically show that CAPSO (Chaotic Accelerated Particle Swarm optimisation) outperforms CPSO (Chaotic Particle Swarm optimisation) for standard optimisation problems. Another method proposed by Pluhacek *et al.* use ensemble learning to obtain the chaotic PSO [2]. This method permits to select the best of six chaotic pseudo random number generators (CPRNG) for each particle. They evaluate their method on the IEEE CEC13' Real-Parameter Single Objective optimisation benchmark set. However, the parameter values of the maps are defined a priori and unchanged during the simulation.

For a given problem, instead of using chaotic dynamics for a system with fixed parameters, Pluhacek *et al.* [11] choose to vary the parameters of the Lozi map [12]. The values of this discrete map are used in an algorithm to

generate a Chaotic Pseudo Random Numbers (CPRNG) in order to replace the Random Number Generator (RNG) used in a PSO algorithm. We recently tune the Lozi map parameter values, as well as the Rössler system parameters to determine the best parameter values for a given optimisation problem: the graph traversal problem [13].

2.2. Metaheuristics for chaotic dynamics

The two previous articles [11, 13] compare all the results for parameters in a given range of values; this approach is time consuming and can be repetitive since the dynamics are the same for various parameter values (see [14] for details on the chaotic dynamics of the Rössler system). To obtain the parameter values of a chaotic system, Senkerik *et al.* [15] propose to use a combination of evolutionary algorithms: Differential Evolution (DE) and Self-Organizing Migrating Algorithm (SOMA). The aim of their study is to obtain faster convergence to specific periodic orbits for a given dynamical system. Gao *et al.* [16] also use Differential Evolution to select periodic orbits by finding the best parameter.

To accelerate the finding of specific parameters of a dynamical system Carbajal-Gómez *et al.* [17] also use Differential Evolution. This method is used to obtain multi-scroll attractors where the positive Lyapunov exponent value is maximised. De La Fraga *et al.* [18] use a multi-objective optimisation algorithm to obtain attractors that maximise the Lyapunov exponent value and minimise the dispersion of the phase space portraits. This optimisation is performed by the well-known multi-objective metaheuristic NSGA-II (non sorting genetic algorithm-II) [19]. Bayesian optimisation has been used once on chaotic dynamics: Abbas *et al.* [20] tune parameters of a chaotic system (Lorenz 95) to underline the capabilities of Bayesian optimisation method on complex system. Indeed Bayesian optimisation has been successfully employed to optimise Machine Learning parameters [21] and multi-level optimisation problems [22].

Based on the aforementioned state-of-the-art analysis, this work is, to the best of our knowledge, the first to use bayesian optimisation to tune the parameters of the Rössler chaotic system.

3. Problem Description

This section presents in detail the problem optimised in this work, i.e. the CACOC mobility model which chaotic parameters will be optimised using bayesian optimisation. Section 3.1 first presents the CACOC model before providing details on the chaotic system parameter to optimise in section 3.2. Finally the coverage metric used to evaluate the performance of the model is introduced in section 3.3.

3.1. Chaos based UAV swarm mobility model

Pheromone-based mobility models [23] have shown promising results to optimise the coverage of an area by a swarm of UAVs. Such models use movement probabilities defined by the pheromone levels as presented in Tab. 1. In order to improve the performance of these approaches, our previous work [3] proposed to combine ACO with the chaotic behaviour of a dynamical system resulting in the CACOC mobility model: Chaotic Ant Colony optimisation for Coverage. More precisely, CACOC replaces the random components of the pheromone-based mobility model by a chaotic behaviour such that the exploration capabilities of the UAV swarm is improved.

Before presenting CACOC in detail, we introduce a simpler mobility model which only includes chaotic dynamics (Chaotic Rössler Mobility Model - CROMM) before introducing CACOC that combines both chaotic dynamics with an Ant Colony exploration algorithm.

Table 1: Pheromone action table for ACO UAV mobility model. $left$ is the amount of pheromone sensed on the left of the UAV, $ahead$ is the amount of pheromone sensed in front of the UAV and $right$ is the amount of pheromone sensed on the right of the UAV; $total = left + ahead + right$.

Probability of action		
Left	Ahead	Right
$p_L = \frac{total - left}{2 \times total}$	$p_A = \frac{total - ahead}{2 \times total}$	$p_R = \frac{total - right}{2 \times total}$

3.1.1. CROMM

In CROMM the UAVs are considered to have a constant speed and choose a movement direction at each discrete time step: A for ahead, R for 45° right and L for 45° left. The next direction choice is given by the first return map (Fig. 1). This map underlines the dynamical signature of the Rössler system giving ρ_{n+1} versus ρ_n. Thus, the next action depends on the previous one:

- if $\rho_n < 1/3$ then direction is right (R);
- if $1/3 \leq \rho_n < 2/3$ then direction is left (L);
- else the direction is ahead (A).

This basic mobility model is named CROMM and its pseudo code is detailed in Alg. 1.

In that case, the good exploration performance of the UAVs is due to the periodic orbits of the Rössler system that lead to patterns. The periodic orbits with low periods are considered as skeleton of the chaotic dynamics and are often visited during the simulation process using a Runge-Kutta (4^{th}

Algorithm 1 CROMM mobility model

procedure CROMM
    current state ← "ahead"
loop:
    $\rho \leftarrow$ *next value in the first return map (Fig. 1)*
    if $\rho < \frac{1}{3}$ **then** *current state* ← "right"
    else if $\rho < \frac{2}{3}$ **then** *current state* ← "left"
    else *current state* ← "ahead"
    end if
    move according to the *current state*
end procedure

order) algorithm. As illustrated in Fig. 1 the period 1 orbit leads to symbols AAAAA... (straight line trajectory); the period 2 leads to the pattern ARARA... (circular trajectory) and the period 4 leads to the serpentine pattern RALARALA... A random initial condition is given to the Rössler system to ensure different trajectories for each UAV of the swarm.

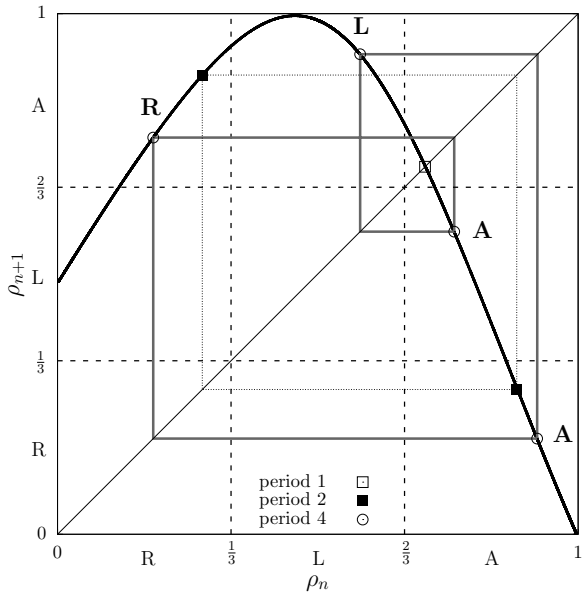

Figure 1: First return map of the Rössler attractor (Figure 2). This map is partitioned in three parts indicating the UAV directions for the CROMM mobility model: L (left), A (ahead) and R (right). Orbits of period 1, 2 and 4 respectively illustrate the patterns A: straight line, AR: turn and RALA: serpentine trajectory transition taking place between the periodic points highlighted in dark grey.

3.1.2. CACOC

CACOC combines a pheromone-based model with chaotic dynamics. If there is no virtual pheromones to guide the UAV (pheromones are deposited by each UAV to indicate areas they already visited), CROMM is used. The UAVs share a map of virtual pheromones that indicates recently visited areas when high pheromone concentrations are present. The UAVs then have a higher probability to move to the least recently visited areas. The pheromones have repulsive properties and the next choice of direction depends on the total amount of pheromones sensed around the UAV. We also used the Rössler first return map values to choose the next direction with the pheromones' perception instead of a random number. Consider that p_L, p_A and p_R are inversely proportional to the total amount of pheromones sensed respectively to the left, ahead and right of the UAV and that $p_R+p_L+p_A = 1$ (Tab. 1). Thus, with ρ_n taken from the first return map (Fig. 1), the next direction is chosen according to these rules:

- if $\rho_n < p_R$ next direction is right;
- if $p_R \leq \rho_n < p_R + p_L$ next direction is left;
- else the direction is ahead.

The pseudo-code of the CACOC algorithm is detailed in Alg. 2.

Algorithm 2 CACOC mobility model

procedure CACOC
    current state ← "ahead"
loop:
    $\rho \leftarrow$ *next value in the first return map (Fig. 1)*
    if no pheromone sensed in the neighbourhood **then**
        current state ← $CROMM(\rho)$ # see Alg. 1
    else
        if $\rho < p_R$ **then** *current state* ← "right" # see Tab. 1 for p_R
        else if $\rho < p_R + p_L$ **then** *current state* ← "left" # see Tab. 1 for p_R and p_L
        else *current state* ← "ahead"
        end if
    end if
    move according to the *current state*
end procedure

The coverage performance of the CACOC mobility model has been empirically demonstrated against the original pheromone based mobility model from Kuiper & Nadjm-Tehrani [23] and mobility models based on other chaotic systems [3].

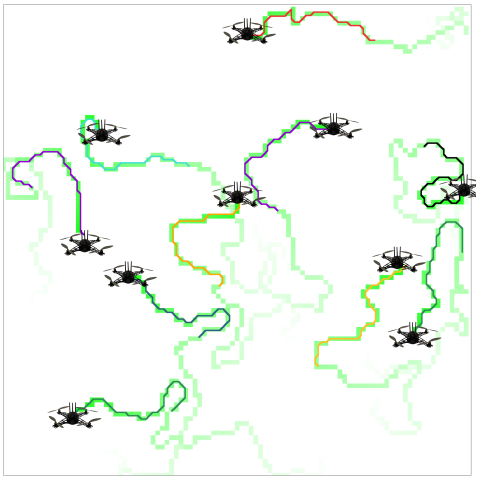

Figure 2: Sample trajectories generated with the CACOC model

3.2. Rössler parameters

CACOC uses chaotic dynamics from the Rössler system which relies on a three differential equations system (see Equation 1). Depending on its parameterisation, the Rössler system can produce periodic, quasi-periodic or chaotic dynamics. The dynamics will therefore impact the mobility model and thus the performance of CACOC.

Studying nonlinear dynamics requires dedicated tools to explore these behaviours. The bifurcation diagram is one tool that displays the solutions in a topological period depending the variation of one selected parameter. It underlines the transitions for parameters leading to periodic or chaotic solutions. We already proposed a detailed analysis of the bifurcation diagram [14] that permits to have the templates (tool to details the topological structure of the chaotic solution) and chaotic mechanisms of attractors solution of the Rössler [4] system for $\alpha \in]-2; 1.8[$

$$\begin{cases} \dot{x} = -y - z \\ \dot{y} = x + ay \\ \dot{z} = b + z(x - c) \ , \end{cases} \quad \text{where} \quad \begin{cases} a = 0.2 + 0.09\alpha \\ b = 0.2 - 0.06\alpha \\ c = 5.7 - 1.18\alpha \ . \end{cases} \tag{1}$$

The reader is referred to [14] for details on the use of the α parametrisation introduced by [24]. Here we focus our study on the attractors with a template with only two branches, that is to say, when $\alpha \in]-0.8; 0.4[$. The upper part of Fig. 5 details the partition of the two branches on the variable y_n and its linear regression depending on α. Thus, for this range of values for α, the first return map to the Poincaré section is only composed of two-branches: an increasing branch and a decreasing branch. However, varying the α parameter will impact the periodic orbits populating the maps which

are used in CACOC [3]. For that reason, this work focuses on optimising the α value of the Rössler system in order to improve CACOC algorithm's area coverage performance detailed in the next section.

3.3. Metric for optimisation: slope of coverage *and* slope of fairness

The performance of the mobility model is evaluated in terms of area coverage [3]. The coverage is the percentage of cells of the area that is visited during the simulation. More precisely, UAVs evolve on a square grid of 100×100. As soon as a UAV attains a cell, it is considered as visited for the whole simulation period. This metric measures the ability of the swarm to quickly perform a first complete scan of the whole area:

$$coverage = \min_{t} \max_{\text{traj}_t(x)} Scan(\text{traj}_t(x)) \tag{2}$$

where t is the number of steps, $x = [(x_1, y_1), \ldots, (x_n, y_n)]$ the position of the n UAVs, $\text{traj}_t(x)$ the trajectories of the UAVs from 0 to t and $Scan$ the percentage of 100×100 area covered based on the trajectories of the UAVs. In this work, we evaluate the first steps of the CACOC model to optimise its initial behaviour. For this purpose the first 500 steps are used to extract the slope of a linear regression $a \times x$. Consequently, CACOC efficiency for initial spreading of the UAVs is optimised using this slope of coverage.

The fairness evaluates if all cells are regularly and equally visited. This is measured by the standard deviation of their respective number of visits [**?**]:

$$fairness = \sqrt{\sum_{c \in C} \frac{(\overline{nbScan_c} - nbScan_c)^2}{|C|}} \tag{3}$$

with $nbScan : C \to N$ the function returning the number of times a given cell has been scanned. To evaluate the fairness during the whole simulation, we perform a linear regression $(a \times x + b)$ using the last 4500 steps. This balances the coverage initial slope that only evaluates the initial UAV trajectories.

In order to obtain meaningful results for these metrics, simulations of the CACOC mobility model must be conducted for a minimum number of UAVs and simulation steps. Such simulations are thus computationally demanding and prevent from using global optimisation methods that require a large number of function evaluations. This motivates our choice of a surrogate-based optimisation approach where the number of objective function calls is limited.

4. Bayesian Opimisation

Bayesian optimisation is a model-based approach which aims at solving black-box or very time-consuming problems. It can be assimilated as an

optimisation algorithm where the formal expression of the objective function may be unknown or very difficult to obtain. To overcome this issue and reduce computation cost, Bayesian optimisation generates a surrogate model of the unknown function using Gaussian processes [6]. It samples promising zones in the feasible region by computing a distribution of the objective function (see Figure. 3). This distribution gives prior knowledge on the location of the optimal solution. Bayesian optimisation is thus characterised by two important mechanisms:

- a probability measure describing prior beliefs on the optimal solution location;
- the acquisition function which allows to gain information on the location of the minimum value of the objective function.

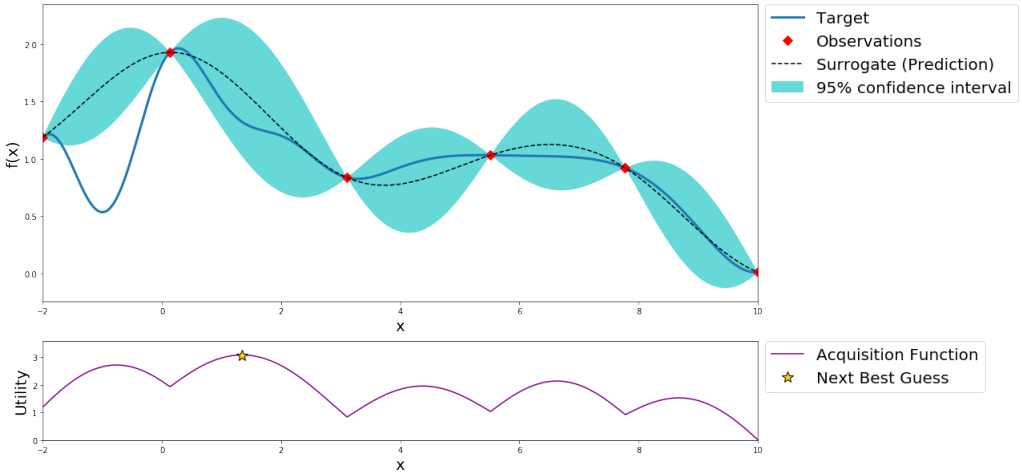

Figure 3: Bayesian optimisation for function approximations

Considering a cost function $F(x)$, Gaussian processes determine the probability distribution of the function $F(x)$ at each x. These distributions are Gaussian and thus characterised by a mean value $\mu(x)$ and a variance $\sigma^2(x)$. Hence a probability distribution over functions can be defined as follows:

$$P(F(x)|x) \sim \mathcal{N}(\mu(x), \sigma^2(x)) \tag{4}$$

The parameters $\mu(x)$ and $\sigma^2(x)$ have to be estimated by fitting the Gaussian processes to the data. Using several observations, we obtain a sample of a multivariate Gaussian distribution [26], determined by a mean vector and a covariance matrix. In fact, Gaussian processes generalise the notion of multivariate Gaussian distribution. For complex nonlinear functions, the covariance matrix is defined using a kernel function $k(x, x')$. This covariance matrix defines the correlation between data. Two distant data x and x'

should not influence each other while two close data are strongly correlated.

$$F(x) \sim \mathcal{GP}(\mu(x), k(x, x')) \tag{5}$$

where $\mathcal{GP}$ stands for Gaussian process. The squared exponential kernel is often used and defined as follows:

$$k(x, x') = l \cdot \exp\left(-\frac{\|x - x'\|^2}{2\sigma_0^2}\right) \tag{6}$$

with parameters l and σ_0^2.

To fit the Gaussian process to the data, the likelihood is optimised from the evaluations of each observation. Each time a new point is added to the model, a re-optimisation is performed to maximise the likelihood. The question is to determine a new point. This is achieved by optimising an acquisition function which statistically models our confidence to find the location of the optimal value. Several acquisition functions exist such as the Maximum Probability of Improvement (MPI), the Expected Improvement (EI), or the Lower-Confidence Bounds (LCB). They are computed as follows:

- $\text{acq}_{MPI}(x) = \Phi(\gamma(x))$.
- $\text{acq}_{EI}(x) = \sigma(x)(\gamma(x)\,\Phi(\gamma(x)) + \phi(\gamma(x)))$.
- $\text{acq}_{LCB}(x) = \mu(x) - k\sigma(x)$.

where $\gamma(x) = \frac{F(x_{best} - \mu(x))}{\sigma(x)}$, Φ is the standard cumulative distribution function, ϕ the standard normal probability density function and k is a parameter allowing to balance exploration-exploitation.

Finally, Alg. 3 depicts the different steps of the standard Bayesian optimisation algorithm. The algorithm is initialised with a random sample of points (line 2). These points are then evaluated with the true but expensive black-box objective function (line 3). At this point, the GP model is fitted to the set of points and their corresponding objective value (line 5). An acquisition function is selected (line 7) and optimised in order to determine the next guess or sample point (line 8). This new point is evaluated with the black-box objective function (line 9) and the GP model is updated accordingly (line 10). The algorithm iterates over the aforementioned steps until convergence.

5. Experiments

This section contains the details about the experimental settings and results. The configuration of the CACOC simulations and of the bayesian optimisation algorithm are given in Section 5.1. Experimental results are then presented and analysed in Section 5.2

Algorithm 3 Bayesian optimisation

```
1: function SOLVE(problem,n,k)
2:     X =initRandom(n);
3:     Y =problem.evaluate(X)
4:     model=GP(X, Y)
5:     model.update()
6:     while not has_converged() do
7:         acq = getAcquisition(k);
8:         x_new =acq.optimise();
9:         y_new =problem.evaluate(x_new);
10:        model.update(x_new,y_new);
11:    end while
12:    return model.best;
13: end function
```

5.1. Experimental Setup

In order to find the best chaotic parameter of the CACOC mobility model that optimises the area coverage performance of a UAV swarm, the framework presented in Figure **??** is used. Solutions provided by the bayesian optimisation algorithm are evaluated using a graph-based UAV swarm simulator in which CACOC has been implemented.The setup used for the UAV swarm simulation and the bayesian optimisation experiments are detailed in the next two sections.

5.1.1. CACOC simulation parameters

The UAV simulation area is a 100 m $\times$ 100 m square, divided in square cells of 1 m $\times$ 1 m. 10 autonomous UAVs are used, all with a constant speed of 1m/s. All depart from a base station located in the middle of the bottom edge of the area, i.e. position (50,0). At each simulation step, each UAV can do one of the following three actions: (1) go ahead: the UAV keeps the same direction; (2) go left: the UAV turns left with a 45° angle; (3) go right: the UAV turns right with -45° angle. To prevent collisions between the UAVs they all have non equal flight altitudes [23]. Each simulation is run for 500 steps. The α parameter of the Rössler system can be set in the interval [-0.8:0.4]. CACOC simulation parameters are summarised in Tab. 2. We manage the border of the area as follows: when the obtained position of the UAV is outside the environment we compute the symmetrical point where the symmetry axis is the border.

5.1.2. Bayesian optimisation parameters

Table 3 describes all parameters used by the Bayesian optimisation algorithm. The initial design consists of 10 randomly selected points in the parameter search space. The covariance matrix has been defined using the

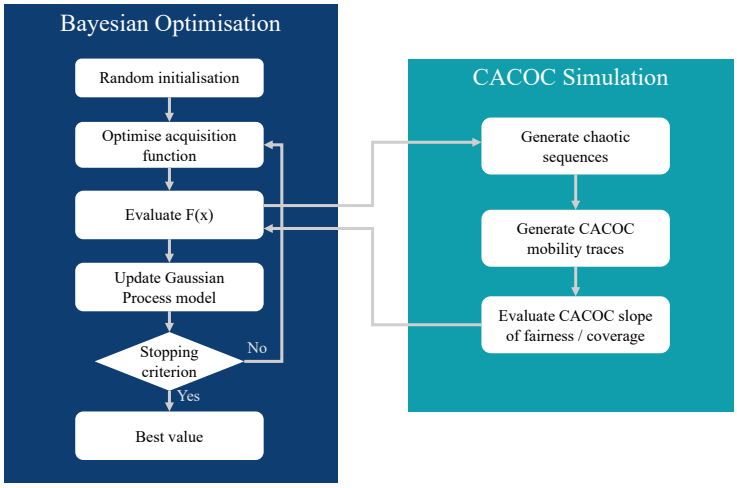

Figure 4: Experimental framework

Table 2: Parameters for the CACOC simulation.

Parameter Name	Parameter Value
Simulation area	
Geographical Area	100 m × 100 m
Number of cells	100 × 100
UAV Autopilot	
UAVs speed	1 m/s
Possible UAV actions	ahead, 45° left, 45° right
Initial UAVs position	middle of the bottom of the map
Experiments	
Number of UAVs	[10]
α range	[-0.8:0.4]
Simulation steps	500

"Mattern" kernel function. This covariance matrix defines the correlation between data. The acquisition function is the Lower-Confidence Bounds (LCB). Since we wish to maximise the coverage, we consider the opposite target value. The optimisation of the acquisition function is done using a local search algorithm which is restarted 20 times. The acquisition function is an auxiliary non-linear function. It can be difficult to find its global optimal solution. Nevertheless, we are not absolutely looking for a global optimal but only for a local optimal that would improve our knowledge about the true objective function. Therefore, a local search approach with multiple restart can provide a "good enough" local optimal solution. The maximum number of iterations (newly added points to the model) has been set to 45. Finally, the Bayesian optimisation stops if the newly added points to the model have an absolute deviation lower than $1.0e^{-5}$. Finally, we did not add any time limit.

Table 3: Experiment parameters for Bayesian optimisation.

Parameters	Value
Initial design	10 random points
Kernel	Mattern
Acquisition	LCB
Exploration weight	2
Acquisition (restart)	20
Maximum iteration	45
Tolerance	$1.0e^{-5}$
Maximum time	None
Independent runs	30

5.2. Experimental Results

Fig. 4 illustrates the main advantage of the Bayesian optimisation to reduce the computation time. Indeed, we can easily observe that the approach acquires points in promising areas without exploring the ones which are statistically less valuable in terms of objective value. Most of the periodic parts of the bifurcation diagram are excluded due to their least performing results in terms of speed of coverage. Thus the algorithm focuses on the most promising areas in the chaotic regions of the bifurcation diagram.

The first lower part of Fig. 5 details the slope of coverage values obtained for the 30 independent runs of the Bayesian optimisation algorithm. All these runs give a slope of coverage value higher than $7.9e^{-4}$. These 30 values are better than the average value of CACOC slope of coverage for $\alpha = -0.25$ that is equal to $7.5e^{-4}$ (see Figure 13 of [3]). On top of that, during the experiments conducted to validate the CACOC approach, we did not obtain any slope of coverage higher than $8e^{-4}$, and in Fig. 5, we mostly obtain

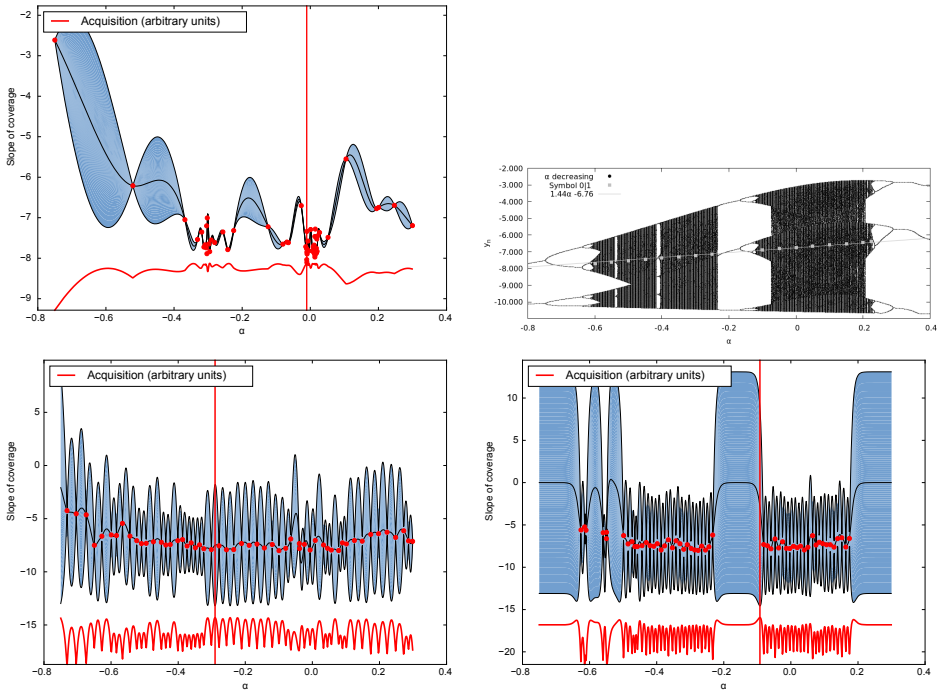

Figure 5: Bifurcation diagram of the Rössler system (periodic region are characterized with lines and chaotic regions are black area illustrating the non periodicity of the chaos). Bayesian optimisation for 3 different runs. The last one shows the interest of the Bayesian optimisation which focuses on chaotic regions of the bifurcation diagram and avoiding periodic ones. The Bayesian optimisation can follow the chaotic region in order to avoid periodic ones where the performance of CACOC are reduced.

values higher than $8e^{-4}$. These values underline the exploration capabilities of the CACOC algorithm by better spreading the UAVs in the environment.

The bottom part of Fig. 5 details the slope of fairness values. The slope of fairness contribute to the evaluation the algorithm on a long term. The values obtained with the Bayesian optimisation are in the range of values already obtained in our previous work $[2.5e^{-3}; 4e^{-3}]$ (see Figure 13 of [3]). Thus the optimisation of the parameter using this metrics do not permit to discriminate significant improvements. For the remainder of this article, we will only consider results of the Bayesian optimisation regarding the *slope of coverage*.

For the 30 runs, the results can be organized in groups depending on their α value:

- 7 values are in $\alpha \in [-0.35; -0.25]$;
- 3 values are in $\alpha \in [-0.12; -0.073]$;
- 20 values are in $\alpha \in [-0.073; 0.042]$.

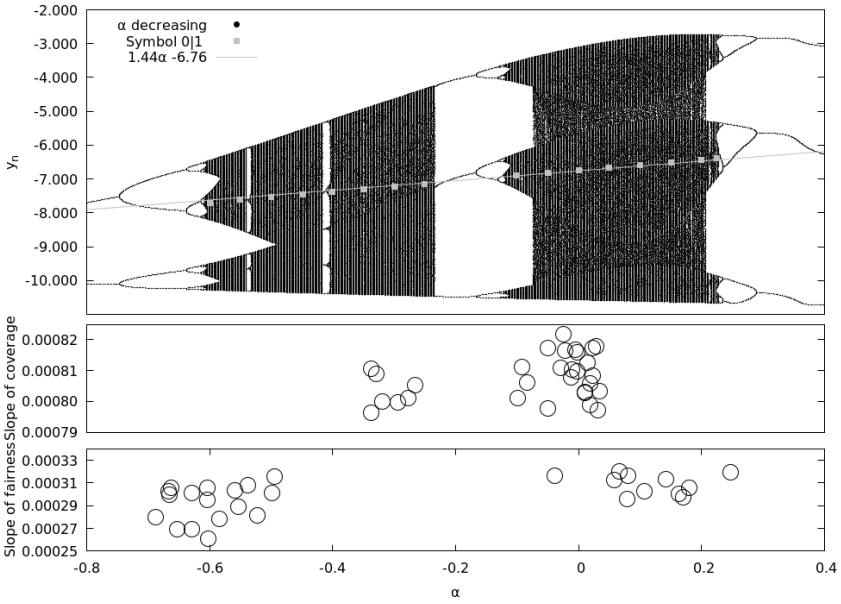

Figure 6: Bifurcation diagram detailing the chaotic and periodic solution depending on the value of α. The partition between the symbols "0" and "1" indicates that the same chaotic mechanism characterise the topological properties of the attractor for these value of α. The values for the slope of coverage and for the fairness are the 30 best values obtain after the Bayesian optimisation. Thus we can briefly conclude that not only the chaotic mechanism is important for CACOC but also the associated orbits it contains. For the slope of coverage, this is due to the absence of best solution out of the range $\alpha \in [-0.4; 0.05]$. For the fairness, values of *alpha* below -0.4 give the best results.

Unfortunately, with the partition of the first return map in three equal portions, the solution with $\alpha \in [-0.12; -0.073]$ cannot be considered as valid because it leads to a regular pattern in terms of mobility model (see Fig. 1 for details on the partition). Thus, we obtain two groups of values giving the best known value of α for CACOC. Also out of these range of values, even for the same chaotic mechanism, CACOC is not able to achieve such good performance in terms of slope of coverage.

From Fig. 5 we can conclude that with $\alpha \in [-0.4; 0.05]$, the specific properties of the orbits constituting the chaotic mechanism are better, even if they have the same chaotic mechanism (two branches). Thus, we can explore the dynamics to deeply understand the properties related to the periodic orbits and the performance of CACOC. In the next section, we will study the dynamical properties of the attractors to details their influence on the performance of the CACOC mobility model.

Table 4: Comparison of the first return map shapes (see Fig. 6 for the first return maps) using $f(0)$ ($\rho_{n+1} = f(\rho_n)$ with $\rho_n = 0$) and their impact in term of transitions allowed for CROMM mobility model (included in the CACOC mobility model).

α	$f(0)$	transitions from R
-0.47	0.67616	A
-0.25	0.48856	A and L
0	0.34562	A and L
0.034	0.33387	A and L
0.1	0.319912	A, L and R

6. Dynamical analysis

In a previous paper [14], we provide a detailed analysis of the topological properties of the attractor solution to the Rössler system. This first paragraph summarises these first steps of the topological characterisation method necessary to obtain comparable first return maps. The chaotic attractors are bounded by a genus–1 torus for the considered values of α. As a consequence, a Poincaré section with only one component is required. Further to the orientation convention introduced in [27], a ρ_n value represents the Poincaré section with an orientation from the inside to the outside.

Further to the results obtained by the Bayesian optimisation, we select values of α. We now present the first return maps of five attractors with different values of α selected to illustrate the dynamics occurring in this bifurcation diagram. First of all, the five first return maps are made with an increasing branch followed by a decreasing branch. However, the initial value of the increasing branch $f(0)$ ($\rho_{n+1} = f(\rho_n)$ with $\rho_n = 0$) is not the same. As illustrated in Tab. 4, the increase of α concords with the increase of the length of the first increasing branch. The latter has an impact on the possible transitions allowed in the CROMM mobility model (by extension, in the CACOC mobility model as well). We remind that the first return maps are equally partitioned using three symbols independently of the branch orientation with L, R and A to give a direction to the UAVs. As a consequence, the problem cannot have better results for values above $\alpha = 0.034$ because these mobility models allow transitions from R to R and this is not profitable (Fig. 6): it leads to small turns (see Tab. 3 in [3] for details). On the other hand, when α is lower than -0.47 transitions from R to L are removed. This also explains why the Bayesian optimisation method does not find better results for α values lower than -0.34: not enough transition from L to R are allowed to obtain an efficient mobility model.

Considering the groups of points giving the best results, the third group with $\alpha \simeq 0$ gives the best results in terms of slope of coverage: it is higher

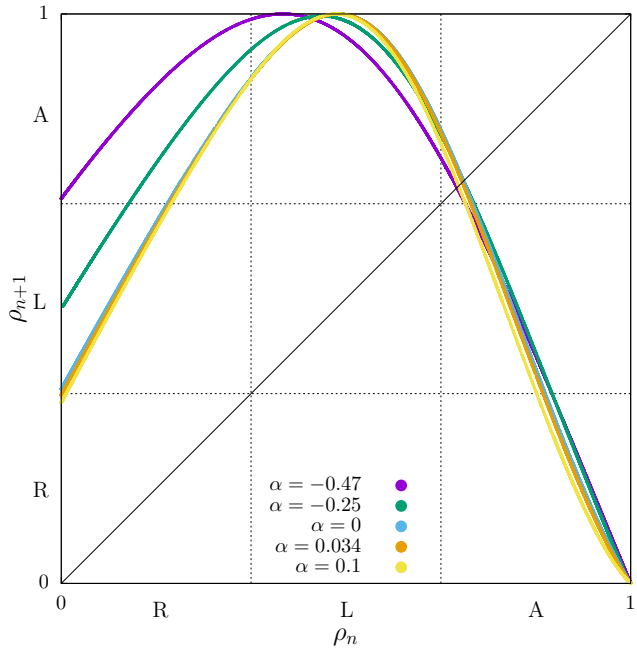

Figure 7: First return maps for α values of Tab. 4. The partition indicate the transitions when there is no pheromones to guide the UAVs (CROMM mobility model used in the CACOC mobility model).

than $8.1e^{-4}$ for nine out of twenty values. We can conclude that better results are obtained when $\alpha = 0.034$. Compared to the first group of points, the chaotic dynamic is the same (same chaotic mechanism described by a first return map with two branches) for $\alpha = -0.25$ and $\alpha = 0.034$ (Fig. 6). However, for the latter the dynamic is developed in a sense that with the first branch going from 1/3 up to 1 leading to supplementary periodic orbits for transition from R to L. This permits to have transitions from R to L and A without transition from R to R as it is the case for $\alpha = 0.1$ (Fig. 6).

7. Conclusion

The CACOC (Chaotic Ant Colony optimisation for Coverage) algorithm has been developed for a swarm of UAVs which purpose is to cover an area. This mobility model uses the Rössler system to introduce chaotic dynamics. In this article, we used Bayesian optimisation to obtain the best parameters for the Rössler system. Our experimentations underline that this method permits to explore efficiently a bifurcation diagram by-passing periodic regions. The results provide two groups of points providing excellent results in terms of slope of coverage for the swarm. Using first return maps, we compare the chaotic dynamics to select the best parameter. This best

solution corresponds to a first return map with a fully developed first branch according to our partition in three equal parts.

For future works, we plan to analyse the performance of the obtained CACOC parameterisation on different area coverage scenarios, e.g. varying the number of UAVs as previously done in [**?**], the size and topology of the area. We also plan to explore more Rössler system dynamics by increasing the number of parameters in our Bayesian optimisation model. We will also consider other dynamical systems leading to chaotic dynamics bounded by a higher genus torus such as Lorenz and Chen systems with techniques used to perform their topological analysis [28].

Acknowledgments

This work relates to Department of Navy award N62909-18-1-2176 issued by the Office of Naval Research. The United States Government has a royalty-free license throughout the world in all copyrightable material contained herein.

The experiments presented in this paper were carried out using the HPC facilities of the University of Luxembourg [29] (see `http://hpc.uni.lu`).

References

References

[1] A. H. Gandomi, G. J. Yun, X.-S. Yang, S. Talatahari, Chaos-enhanced accelerated particle swarm optimization, Communications in Nonlinear Science and Numerical Simulation 18 (2) (2013) 327–340. doi:10.1016/j.cnsns.2012.07.017.

[2] M. Pluhacek, R. Senkerik, D. Davendra, Chaos particle swarm optimization with Eensemble of chaotic systems, Swarm and Evolutionary Computation 25 (2015) 29–35. doi:10.1016/j.swevo.2015.10.008.

[3] M. Rosalie, G. Danoy, S. Chaumette, P. Bouvry, Chaos-enhanced mobility models for multilevel swarms of UAVs, Swarm and Evolutionary Computation 41 (2018) 36–38. doi:10.1016/j.swevo.2018.01.002.

[4] O. Rössler, An equation for continuous chaos, Physics Letters A 57 (5) (1976) 397–398. doi:10.1016/0375-9601(76)90101-8.

[5] G. Diaz, A. Fokoue-Nkoutche, G. Nannicini, H. Samulowitz, An effective algorithm for hyperparameter optimization of neural networks, IBM Journal of Research and Development 61 (4) (2017) 9:1–9:11. doi:10.1147/JRD.2017.2709578.

[6] J. Mockus, Bayesian approach to global optimization: theory and applications, Vol. 37, Springer Science & Business Media, 2012.

[7] K. Tatsumi, Y. Obita, T. Tanino, Chaos generator exploiting a gradient model with sinusoidal perturbations for global optimization, Chaos, Solitons & Fractals 42 (3) (2009) 1705–1723. doi:10.1016/j.chaos.2009.03.088.

[8] I. Tokuda, K. Onodera, R. Tokunaga, K. Aihara, T. Nagashima, Global bifurcation scenario for chaotic dynamical systems that solve optimization problems and analysis of their optimization capability, Electronics and Communications in Japan (Part III: Fundamental Electronic Science) 81 (2) (1998) 1–12. doi:10.1002/(sici)1520-6440(199802)81:2<1::aid-ecjc1>3.0.co;2-t.

[9] K. Tatsumi, T. Tanino, A perturbation based chaotic system exploiting the quasi-newton method for global optimization, International Journal of Bifurcation and Chaos 27 (04) (2017) 1750047. doi:10.1142/s021812741750047x.

[10] D. H. Wolpert, W. G. Macready, No free lunch theorems for optimization, IEEE Transactions on Evolutionary Computation 1 (1) (1997) 67–82. doi:10.1109/4235.585893.

[11] M. Pluhacek, R. Senkerik, I. Zelinka, PSO algorithm enhanced with lozi chaotic map - tuning experiment, AIP Publishing LLC, 2015, p. 550022. doi:10.1063/1.4912777.

[12] R. Lozi, Un attracteur étrange (?) du type attracteur de Hénon, Le Journal de Physique Colloques 39 (C5) (1978) C5–9.

[13] B. Changaival, M. Rosalie, G. Danoy, K. Lavangnananda, P. Bouvry, Chaotic traversal (CHAT): Very large graphs traversal using chaotic dynamics, International Journal of Bifurcation and Chaos 27 (14) (2017) 1750215. doi:10.1142/s0218127417502157.

[14] M. Rosalie, Templates and subtemplates of Rössler attractors from a bifurcation diagram, Journal of Physics A: Mathematical and Theoretical 49 (31) (2016) 315101. doi:10.1088/1751-8113/49/31/315101.

[15] R. Senkerik, I. Zelinka, D. Davendra, Z. Oplatkova, Utilization of SOMA and differential evolution for robust stabilization of chaotic logistic equation, Computers & Mathematics with Applications 60 (4) (2010) 1026–1037. doi:10.1016/j.camwa.2010.03.059.

[16] F. Gao, Y. B. Qi, Q. Yin, J. Q. Xiao, Solving problems in chaos control though an differential evolution algorithm with region zooming, Applied Mechanics and Materials 110-116 (2011) 5048–5056. doi:10.4028/www.scientific.net/amm.110-116.5048.

[17] V. Carbajal-Gómez, E. Tlelo-Cuautle, F. Fernández, Optimizing the positive lyapunov exponent in multi-scroll chaotic oscillators with differential evolution algorithm, Applied Mathematics and Computation 219 (15) (2013) 8163–8168. doi:10.1016/j.amc.2013.01.072.

[18] L. G. de la Fraga, E. Tlelo-Cuautle, Optimizing the maximum lyapunov exponent and phase space portraits in multi-scroll chaotic oscillators, Nonlinear Dynamics 76 (2) (2014) 1503–1515. doi:10.1007/s11071-013-1224-x.

[19] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197. doi:10.1109/4235.996017.

[20] M. Abbas, A. Ilin, A. Solonen, J. Hakkarainen, E. Oja, H. Jarvinen, Empirical evaluation of Bayesian Optimization in parametric tuning of chaotic systems, International Journal for Uncertainty Quantification 6 (6) (2016) 467–485. doi:10.1615/int.j.uncertaintyquantification.2016016645.

[21] T. T. Joy, S. Rana, S. Gupta, S. Venkatesh, Hyperparameter tuning for big data using bayesian optimisation, in: 2016 23rd International Conference on Pattern Recognition (ICPR), 2016, pp. 2574–2579. doi:10.1109/ICPR.2016.7900023.

[22] E. Kieffer, G. Danoy, P. Bouvry, A. Nagih, Bayesian optimization approach of general bi-level problems, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO '17, ACM, New York, NY, USA, 2017, pp. 1614–1621. doi:10.1145/3067695.3082537.

[23] E. Kuiper, S. Nadjm-Tehrani, Mobility models for uav group reconnaissance applications, in: 2006 International Conference on Wireless and Mobile Communications (ICWMC'06), 2006, pp. 33–33. doi:10.1109/ICWMC.2006.63.

[24] J. C. Sprott, C. Li, Asymmetric bistability in the Rössler system, Acta Physica Polonica B 48 (1) (2017) 97. doi:10.5506/APhysPolB.48.97.

[25] J. Schleich, A. Panchapakesan, G. Danoy, P. Bouvry, UAV fleet area coverage with network connectivity constraint, in: Proc. of International Symposium on Mobility Management and Wireless Access - MobiWac'13, Association for Computing Machinery (ACM), 2013. doi:10.1145/2508222.2508225.

[26] C. Rasmussen, C. Williams, Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning), The MIT Press, 2005.

[27] M. Rosalie, C. Letellier, Systematic template extraction from chaotic attractors: I. genus-one attractors with an inversion symmetry, Journal of Physics A: Mathematical and Theoretical 46 (37) (2013) 375101. doi:10.1088/1751-8113/46/37/375101.

[28] M. Rosalie, C. Letellier, Toward a general procedure for extracting templates from chaotic attractors bounded by high genus torus, International Journal of Bifurcation and Chaos 24 (04) (2014) 1450045. doi:10.1142/s021812741450045x.

[29] S. Varrette, P. Bouvry, H. Cartiaux, F. Georgatos, Management of an academic HPC cluster: The UL experience, in: Proc. of IEEE International Conference on High Performance Computing & Simulation (HPCS), 2014. doi:10.1109/HPCSim.2014.6903792.