AUTOMATED TUNING OF A CELLULAR AUTOMATA USING PARALLEL ASYNCHRONOUS PARTICLE SWARM OPTIMISATION

Christoph Tholen, Tarek El-Mihoub, Jan Dierks, Lars Nolle and Alexandra Burger
Autonomous Systems Research Group
Department of Engineering Science
Jade University of Applied Science
Friedrich-Paffrath-Straße 101
26389 Wilhelmshaven, Germany
Email: {christoph.tholen | tarek.el-mihoub | jan.dierks | lars.nolle | Alexandra.burger}@jade-hs.de

Oliver Zielinski
Institute for Chemistry and Biology of the Marine Environment
Carl von Ossietzky University of Oldenburg
Schleusenstraße 1
26382 Wilhelmshaven, Germany
and
DFKI RG Marine Perception
Marie-Curie-Str. 1
26129 Oldenburg, Germany
Email: oliver.zielinski@uol.de

KEYWORDS
SGD, coastal recirculation, cellular automata, optimisation, Parallel Asynchronous Particle Swarm Optimisation.

ABSTRACT

The long term goal of this research is the development of a distributed autonomous low-cost platform for marine exploration. One application of such a platform could be the search for Submarine Groundwater Discharges (SGD) in a coastal environment. In order to design and to test new search strategies for such a platform, a simulation that effectively models the diffusion of groundwater discharge in shallow coastal waters is required. The simulation allows the evaluation of new search strategies without running the risk of losing expensive hardware during the field testing.

In this paper a simulation based on cellular automata was adapted in order to resemble the behaviour of an existing physical model of a SGD. To speed up the optimisation process, a novel adaptation of the Parallel Asynchronous Particle Swarm Optimisation (PAPSO) algorithm was proposed.

Experiments showed that the novel PAPSO was able to reduce the time needed for optimisation by 69.1 %. Furthermore, the results found by PAPSO are 2.1 % better than the results of the Parallel Synchronous Particle Swarm Optimisation (PSPSO) algorithm.

INTRODUCTION

The long term goal of this research is the development of a distributed and autonomous low-cost platform for marine observation. The observatory consists of an autonomous surface vehicle and a small swarm of autonomous underwater vehicles (AUV). Such a system could be used for locating submarine sources of interest, e.g. dumped waste or lost harmful cargo. The problem at hand is the localisation of submarine groundwater discharges (SGD) in coastal waters. SGD consist of an inflow of fresh groundwater and recirculated seawater from the sea floor into the ocean (Figure 1) (Moore 2010). The freshwater that flows into the ocean represents a continuous and significant source of nutrients for the costal marine environment (Nelson et al., 2015).

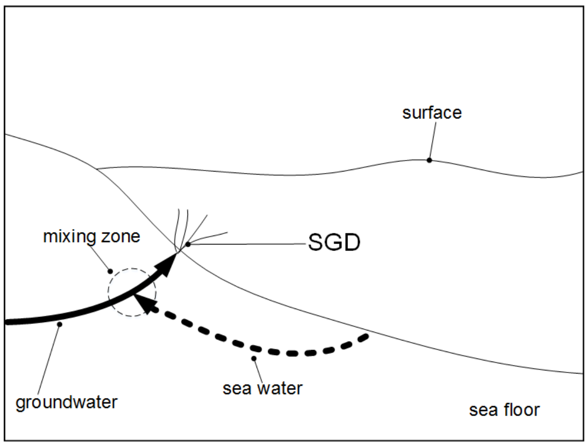

Figure 1: Submarine Groundwater Discharge of Fresh- and Recirculating-Water, modified after Evans and Wilson (2016)

Marine scientists are interested in analysing these discharges, because the nutrients discharged by SGD have a significant influence on the marine ecosystem (Dugan et al., 2010; Beck et al., 2017; Kröncke et al., 2018). However, due to the harsh environmental conditions, locating and analysing SGD requires technical effort (Nelson et al., 2015). Different tracers, including chemical substances, for example Radon (Santos et al., 2009; Kelly et al., 2018), Fluorescent Dissolved Organic Matter (FDOM) (Nelson et al., 2015), conductivity or temperature (Kelly et al., 2018; Röper et al., 2014) can be used to locate SGDs in coastal environments. In this research the temperature was chosen to trace the distribution of the inflow via SGD.

To guide the observatory towards the points of interest, a search strategy is needed. In this project the strategy will be based on Artificial Intelligence methods (Tholen et al., 2018; Tholen & Nolle, 2017). To fine-tune the search algorithms developed, many test dives need to be conducted. To reduce the costs of development and the risk of losing expensive hardware, computer simulations should be used during the design phase. Previously, a simulation based on a cellular automaton was developed (Tholen et al., 2017).

In this paper, this simulation was adapted in order to resemble the behaviour of a physical SGD model. For this adaptation, real measurement data gained from the physical model was used together with Parallel Asynchronous Particle Swarm Optimisation (PAPSO) (Kennedy & Eberhart, 1995; Koh et al., 2006).

SIMULATION

The simulation used here is based on cellular automata (CA). This section introduces the basic concepts of CA and introduces the rules developed during this research.

Cellular Automata

Cellular automata (CA) are mathematical models used for the simulation of complex systems. A CA discretises a system in space and time (Wolfram, 1984). It consists of a finite collection of identical cells. Each cell has a current state, which is updated after each time step and is based on its own previous state and the previous state of its neighbours. The neighbours are defined by the chosen neighbourhood scheme. The most popular schemes are the Moore-neighbourhood and the von-Neumann-neighbourhood. Figure 2 shows an example of a two-dimensional cellular automaton with the dimensions 3 x 3. It uses the von-Neumann-neighbourhood, where the black cell's state depends on its neighbour´s (grey cells) states.

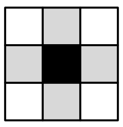

Figure 2: Cellular Automaton of Size 3 x 3

The implemented simulation environment covers an area of 200 cm x 200 cm. This area is divided into a number of symmetric cells. Each cell has an x- and a y-position as well as a depth and a temperature.

The interaction of cells with their neighbours is based on a set of application specific rules. These rules define the dynamic behaviour of the model (Wolfram, 1984; Tholen et al., 2017).

The developed CA is based on a single simple rule only; the temperature of a cell *x* at time step *t*+1 is calculated as the weighted average of the temperatures of the cell *x* and its neighbouring cells y_n and, if present, the temperature of an existing discharge at the position of the cell *x*. All values will be multiplied by the volume of the cells or by the volume flow rate of the spring respectively. The sum will be divided by the sum of the volume of all cells in the neighbourhood. The rule is given in Equation (1). Here *I* is representing the temperature in °C and *V* is representing the volume of the cells in m^3.

$$I_x^{t+1} = \frac{I_x^t * V_x^t + \sum\left(I_y^t * V_y^t\right) + I_s^t * V_s^t}{V_x^t + \sum\left(V_y^t\right) + V_s^t} \qquad (1)$$

Where:
I_x^{t+1}: temperature of cell *x* in iteration *t+1*
I_x^t: temperature of cell *x* in iteration *t*
V_x^t: Volume of cell *x* in iteration *t*
I_y^t: temperature of neighbour cell y_n in iteration *t*
V_y^t: Volume of neighbour cell y_n in iteration *t*
I_S^t: temperature of spring located at cell *x* in iteration *t*
V_S^t: Volume flow of spring located at cell *x* in iteration *t*

In addition to the rule, the behaviour of a CA also depends on the chosen neighbourhood and cell shape (Tholen et al. 2017).

PHYSICAL SGD MODEL

The physical model used in this research was developed at Jade University. It consist of a basin with dimensions 200 cm x 200 cm x 20 cm, a peristaltic pump and a heat exchanger to simulate the inflow of an SGD. The position of the SGD is located at position (*x* = 40 cm, *y*=130 cm). 36 temperature sensors are installed at the bottom of the basin. In addition, an infrared camera is mounted on the top to measure the surface temperature of the water. A second pump and a vent are installed to enable the simulation of tides and currents. Figure 3 shows a picture of the physical SGD model containing the infrared camera at the top, a shielding against disruptive infrared radiation from the environment and the SGD composed of the peristaltic pump and the heat exchanger.

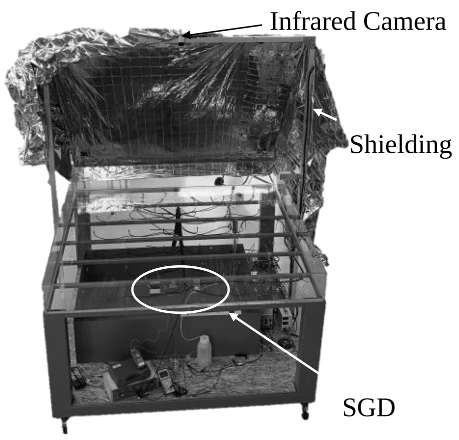

Figure 3: Physical SGD Model with mounted Infrared Camera and Shielding against disruptive Infrared Radiation from the Environment

PARTICLE SWARM OPTIMISATION

PSO is modelled on the behaviour of collaborative real world entities (particles), for example fish schools or flocks of birds, which works together to achieve a common goal (Kennedy & Eberhart, 1995). Each individual of the swarm searches for itself. However, the other swarm members also influence the search behaviour of each individual.

In the beginning of a search, each particle of the swarm starts at a random position and a randomly chosen velocity for each direction of the n-dimensional search space. Then, the particles move through the search space with an adjustable velocity. The velocity of a particle is based on its current fitness value, the best solution found so far by the particle (cognitive knowledge) and the best solution found so far by the whole swarm (social knowledge) (2):

$$\vec{v}_{i+1} = \vec{v}_i \omega + r_1 c_1 (\vec{p}_b - \vec{p}_i) + r_2 c_2 (\vec{g}_b - \vec{p}_i) \quad (2)$$

Where:

$\vec{v}_{i+1}$:	new velocity of a particle,
$\vec{v}_i$:	current velocity of a particle,
ω:	inertia weight,
c_1:	cognitive scaling factor,
c_2:	social scaling factor,
r_1:	random number from range [0,1],
r_2:	random number from range [0,1],
$\vec{p}_i$:	current position of a particle,
$\vec{p}_{best}$:	best known position of a particle,
$\vec{g}_{best}$:	best known position of the swarm.

After calculating the new velocity of the particle, the new position $\vec{p}_{i+1}$ can be calculated as follows:

$$\vec{p}_{i+1} = \vec{p}_i + \vec{v}_{i+1} \Delta t \quad (3)$$

Where:

$\vec{p}_{i+1}$:	new position of a particle,
$\vec{p}_i$:	current position of a particle,
$\vec{v}_{i+1}$:	new velocity of a particle,
Δt:	time step (one unit).

In (3) Δt, which always has the constant value of one unit, is multiplied to the velocity vector $\vec{v}_{i+1}$ in order to get consistency in the physical units (Nolle, 2015). In this research the control parameter values for all experiments were chosen as follows (Eberhardt & Shi, 2000):

ω =	0.729,
c_1 =	1.49,
c_2 =	1.49.

The values of the velocity vector $\vec{v}_0$ will be initialised to zero, to speed up the search of the PSO (Engelbrecht, 2012).

Parallel Asynchronous Particle Swarm Optimisation

Due to the high computational costs needed for the optimisation process of real world problems, different parallel adaptations of the PSO were proposed in the past (Schutte et al., 2004; Koh et al., 2006). Parallel optimisation algorithms can be divided into synchronous and asynchronous algorithms (Koh et al., 2006).

The majority of parallel algorithms proposed in the literature use a synchronous software architecture (Koh et al., 2006). However a great disadvantage of using synchronous optimisation algorithms is the need of balancing the workload of all workers, to avoid idle states of workers. This cannot be guaranteed, if the time for fitness evaluation depends on the input vector of the fitness function (Koh et al., 2006).

The version of the Parallel Asynchronous Particle Swarm Optimisation (PAPSO) introduced by Koh et al. (2006) avoid the disadvantages of the synchronous version of the PSO. This PAPSO algorithm follows the master-slave principle. The master thread containing a queue of all particles ready for evaluation. Also the master performs the decision making process, i.e. calculating the next position, of all particles. The master assigns the first particle, i.e. candidate solution, in the queue to a free thread, i.e. a slave. The slave does the evaluation of the fitness function and returns the fitness value to the master. The master checks if the returned value is better than the personal best of the particle or the global best. If that is the case, the master will update the appropriate value. Subsequently, the master assigns the next particle in the queue to the slave. The task-queue shall ensure that all particles will perform approximately the same number of function evaluations (Koh et al., 2006).

In this work a different implementation of the PAPSO is introduced. This novel implementation does not use the master-slave principle. Instead, all particles are implemented as independent workers. The individual particles compete for the computing resources available. If a particle finds a new best position, it shares the information with the other particles.

EXPERIMENTS

In the first step, the physical model was allowed to run for approximately three hours with constant inflow rate and inflow temperature. During this time the infrared camera at the top took one picture of the surface temperature per minute. These pictures were stored for post processing. From each picture 25 positions were selected and the temperature profile of this points over time was used for the optimisation of the simulation. Figure 4 shows the position of the measurement points (circles) selected for the optimisation process as well as the position of the SGD (diamond shape).

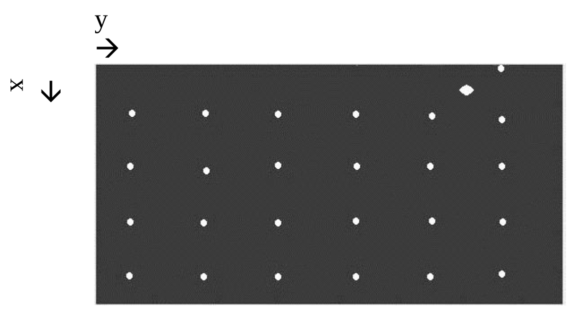

Figure 4: Selected Points for Optimisation and Position of the SGD (Diamond)

In the next step, the simulation based on a CA was optimised using PSPSO, PAPSO, and the temperature profiles of the measurement points. To adapt the behaviour of the CA five different design parameters were defined. The first parameter x_1 is the size of the cells. The value of the cell size can be set to each value of the range $1\text{cm} \leq x_1 \leq 100$ cm. The second design parameter x_2 define the cell shape and the neighbourhood. The value of the design parameter x_2 can be set to integer values in the range $1 \leq x_2 \leq 4$. Table 1 provides maps the different values for the design variable x_2. The third parameter x_3 represents the inflow rate of the SGD in arbitrary units. The value of the parameter can be set to any value of the range $0.01 \leq x_3 \leq 100$. The parameter x_4 represents the inflow temperature of the SGD in °C. The value of the temperature can be set to any value of the interval 20.5 °C $\leq x_4 \leq$ 90 °C. The last design parameter represents the time base of the simulation, i.e. the time per iteration in milliseconds. The value of this parameter can be set to any integer from the interval 500 ms $\leq x_5 \leq$ 60000 ms, with the constraint x_5 modulo 60000 ms = 0.

Table 1: Lookup Table for Design Variable x_2

Value	Meaning
1	Moore neighbourhood
2	Von-Neumann neighbourhood
3	Hexagon shaped cell same perimeter length
4	Hexagon shaped cell same area covered

The following cost function was used for the optimisation of the cellular automaton:

$$\text{fitness} = \sqrt{\left[\sum_{t=0}^{tmax}\left(\sum_{p=1}^{pmax}\left(x_{p,t} - x'_{p,t}\right)^2\right)\right]} \quad (4)$$

Where:

$x_{p,t}$:	Temperature value of the physical model at position p and time t,
$x'_{p,t}$:	Temperature value of the simulation at position p and time t,
$pmax$:	Number of measurement points (25),
$tmax$:	Total time of experiment in minutes (168).

The interaction between the different components of the optimisation process can be depicted from Figure 5. The fitness function represents the key part of the optimisation process. During the fitness evaluation, the measurement data from the physical SGD model are compared with the simulated temperature data generated by the CA (Equation 4). The output of the fitness function, i.e. the fitness for a specific set of design parameters, is used by the optimiser to generate new suitable values for the design parameters. The descried interaction represents one iteration for one particle.

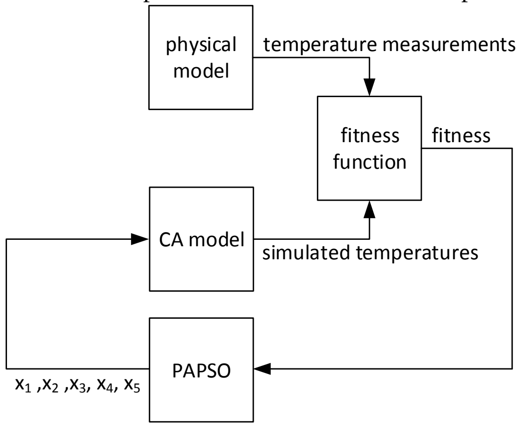

Figure 5: Interaction between the different components (physical model, CA and PAPSO) for the optimisation

It is assumed that the selected values of the design parameters will have a huge impact on the computing time required for fitness function evaluation. To test this assumption, experiments were carried out were only one design parameter was changed at a time, whilst the others were kept constant. Figure 6 depicts the normalised computing time needed to evaluate the fitness function, using the specific input values of the design parameters. It can be observed from Figure 6, that the computing time needed for fitness evaluation depends on the cell size (x_1), the cell shape and neighbourhood (x_2) and the time per iteration (x_5). The computing time needed for the fitness evaluation is independent from the inflow rate (x_3) and the inflow temperature (x_4).

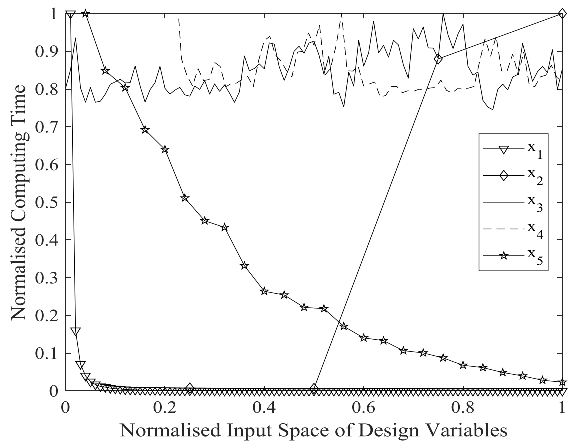

Figure 6: Normalised Computing Time for the different Design Variables

To compare the performance of PSPSO and the novel PAPSO, both algorithms were allowed to run for 200 iterations with using different numbers of particles. The total time needed for the optimisation process and the g_{best} value over time as well as the parameter set of the best position are compared for the different settings.

RESULTS

Figure 7 shows the change of the g_{best} value over time for PAPSO and PSPSO for different swarm sizes. It can be obtained from Figure 7 that the novel PAPSO outperforms PSPSO in terms of time needed for the optimisation and in terms of best fitness values found by the swarm.

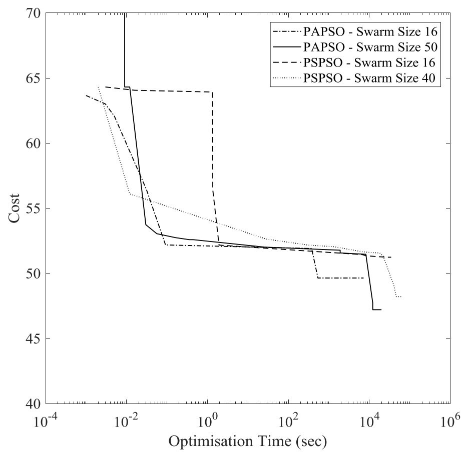

Figure 7: Cost as Function of Optimisation Time (sec) for PAPSO and PSPSO with different Swarm Sizes

Table 2 presents the values of the five design parameters for the best solution found by the PAPSO with a swarm size of 50 particles. The fitness value of this solution was g_{best} = 47.21 (arbitrary units).

Table 2: Optimal Values for the Design Parameters found by PAPSO

Design Parameter	Optimal Value
x_1	37.6 cm
x_2	Hexagon shaped cell same perimeter length
x_3	50.16
x_4	47.72
x_5	600 ms/iteration

To compare the performance of the best solutions found by the PSPSO and the PAPSO, the simulation was set up using the optimal values for the design parameters found by the optimisers. During the run the Root Mean Square Error (RMS) over the time was measured following Equation 5:

$$\text{RMS} = \frac{\sqrt{\left(\sum_{p=1}^{pmax}\left(x_{p,t} - x'_{p,t}\right)^2\right)}}{pmax} \quad (5)$$

Where:

$x_{p,t}$:	Temperature value of the physical model at position p and time t,
$x'_{p,t}$:	Temperature value of the simulation at position p and time t,
$pmax$:	Number of measurement points (25).

Figure 8 shows the RMS errors of the solutions found by PSPSO and PAPSO. It can be seen from the figure that the RMS of the solution found by PAPSO is slightly better than the RMS of the solution found by PSPSO.

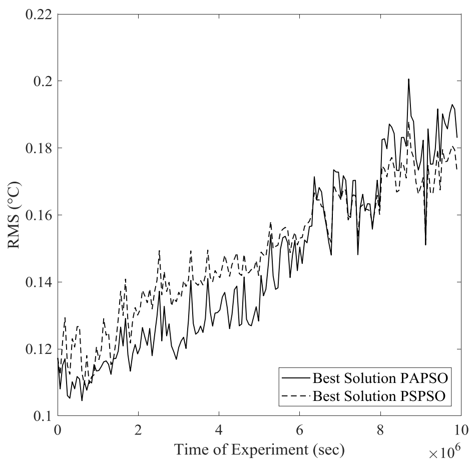

Figure 8: RMS over Time for the Best Solutions, found by PSPSO and PAPSO using a Swarm size of 50 Particles

Table 3 summarises the simulation results. It can be seen from the Table that the performance of the PAPSO with a swarm size of 50 particles is approximately 3.2 % better, than the performance of the PSPSO with the same swarm size. However for this swarm size, the PAPSO operates 6.4 times faster than the PSPSO.

Table 3: Performance of PSPSO and PAPSO for different Swarm Sizes

Algorithm	Time for Optimisation (sec)	gbest (arbitrary units)
PAPSO – Swarm 16	7,228	49.65
PAPSO – Swarm 50	19,967	47.21
PSPSO – Swarm 16	34,708	51.24
PSPSO – Swarm 50	64,572	48.21

CONCLUSION AND FUTURE WORK

In this research, a computer simulation based on cellular automata was optimised, in order to resemble the behaviour of a physical SGD model. To speed up the optimisation process, a novel adaptation of the Parallel

Asynchronous Particle Swarm Optimisation (PAPSO) algorithm was proposed.

It was shown, during experiments that the novel adaptation of the PAPSO was able to reduce the time needed for optimisation, i.e. the optimisation process took only 30.9 % of the optimisation time of the PSPSO. Furthermore, the results found by the PAPSO were slightly better, i.e. 2.1 %, than the results of the Parallel Synchronous Particle Swarm Optimisation (PSPSO) algorithm.

In the next phase of this research, developed search strategies for a single AUV (Tholen et al., 2018) and for a small swarm of AUVs (Tholen & Nolle, 2017) will be tested using the tuned CA. Moreover, the performance of the PAPSO will be investigated further in future research.

REFERENCES

Beck, M. et al., 2017. The drivers of biogeochemistry in beach ecosystems: A cross-shore transect from the dunes to the low water-line. *Mar Chem 190*, pp. 35-50.

Dugan, J. et al., 2010. Give Beach Ecosystems Their Day in the Sun. *Science, Vol. 329, ISSUE. 5996*, p. 1146.

Engelbrecht, A., 2012. Particle Swarm Optimization: Velocity Initialization. *2012 IEEE Congress on Evolutionary Computation*,, pp. 1-8.

Evans, T. & Wilson, A., 2016. Groundwater transport and the freshwater–saltwater interface below sandy beaches. *Journal of Hydrology, 538*, p. 563–573.

Kelly, J. L., Dulai, H., Glenn, C. R. & Lucey, P. G., 2018. Integration of aerial infrared thermography and in situ radon-222 to investigate submarine groundwater discharge to Pearl Harbor, Hawaii, USA. *Limnology and Oceanography. 9999* , pp. 1-20.

Kennedy, J. & Eberhart, R., 1995. Particle swarm optimization. *IEEE International Conference on: Neural Networks, Vol. 4*, pp. 1942-1948.

Koh, B., George, A. D., Haftka, R. T. & Fregly, B. J., 2006. Parallel asynchronous particle swarm optimization. *International Journal for Numerical Methods in Engineering (67)*, pp. 578-595.

Kröncke, I. et al., 2018. Near- and Offshore Macrofauna Communities and Their Physical Environment in a South-Eastern North Sea Sandy Beach System. *Front. Mar. Sci. 5:497*.

Moore, W., 2010. The effect of submarine groundwater discharge on the ocean. *Annual Review of Marine Science, Vol. 2*, pp. 59-88.

Nelson, C. et al., 2015. Fluorescent dissolved organic matter as a multivariate biogeochemical tracer of submarine groundwater discharge in coral reef ecosystems. *Marine Chemistry, 177*, p. 232–243.

Nolle, L., 2015. On a search strategy for collaborating autonomous underwater vehicles. *Proceedings of Mendel 2015, 21st International Conference on Soft Computing*, pp. 159-164.

Röper, T., Greskowiak, J. & Massmann, G., 2014. Detecting Small Groundwater Discharge Springs Using Handheld Thermal Infrared Imagery. *Groundwater (52)*, pp. 936-942.

Santos, I. et al., 2009. Land or ocean? Assessing the driving forces of submarine groundwater discharge at a coastal site in the Gulf of Mexico. *Journal of Geophysical Research: Oceans (114)*, p. 2156–2202.

Schutte, J. F. et al., 2004. Parallel global optimization with the particle swarm algorithm. *Int. J. Numer. Meth. Engng., 61*, pp. 2296-2315.

Tholen, C., El-Mihoub, T., Nolle, L. & Zielinski, O., 2018. *On the robustness of self-adaptive Levy-flight.* Kobe (Japan), IEEE.

Tholen, C. & Nolle, L., 2017. Parameter Search for a Small Swarm of AUVs Using Particle Swarm Optimisation. *Artificial Intelligence XXXIV. SGAI 2017. Lecture Notes in Computer Science (10630)*, pp. 384-396.

Tholen, C., Nolle, L. & Zielinski, O., 2017. On The Effect Of Neighborhood Schemes And Cell Shape On The Behaviour Of Cellular Automata Applied To The Simulation Of Submarine Groundwater Discharge. *Proceedings of the 31st European Conference on Modelling and Simulation (ECMS)*, pp. 255-261.

Wolfram, S., 1984. Universality and complexity in cellular automata. *Physica D: Nonlinear Phenomena, Vol. 10, Issue 1-2*,, pp. 1-35.

AUTHOR BIOGRAPHIES

CHRISTOPH THOLEN graduated from the Jade University of Applied Science in Wilhelmshaven, Germany, with a Master degree in Mechanical Engineering in 2015. Since 2016 he is a research fellow at the Jade University of Applied Science in a joint project of the Jade University of Applied Science and the Institute for Chemistry and Biology of the Marine Environment (ICBM), at the Carl von Ossietzky University of Oldenburg for the development of a low cost and intelligent environmental observatory.

JAN DIERKS graduated from the Jade University of Applied Science with a Bachelor degree in Mechatronics in 2017. He joined Jade University of Applied Sciences as a research fellow in 2017. Currently he is studying his Master degree in Electronics at Jade University of Applied Sciences.

TAREK A. EL-MIHOUB graduated with a BSc in computer engineering from University of Tripoli, Tripoli, Libya. He obtained his MSc in engineering multimedia and his PhD in computational intelligence from Nottingham Trent University in the UK. He was an assistant professor at the Department of Computer Engineering, University of Tripoli. He is currently a postdoctoral researcher with Jade University of Applied Science. His current research is in the fields of applied computational intelligence and autonomous underwater vehicles.

LARS NOLLE graduated from the University of Applied Science and Arts in Hanover, Germany, with a degree in Computer Science and Electronics. He obtained a PgD in Software and Systems Security and an MSc in Software Engineering from the University of Oxford as well as an MSc in Computing and a PhD in Applied Computational Intelligence from The Open University. He worked in the software industry before joining The Open University as a Research Fellow. He later became a Senior Lecturer in Computing at Nottingham Trent University and is now a Professor of Applied Computer Science at Jade University of Applied Sciences. His main research interests are computational optimisation methods for real-world scientific and engineering applications.

ALEXANDRA BURGER graduated from the University of Applied Science in Schweinfurt with a degree in Electronics. She obtained a degree of automation engineering and a Ph.D. from University of Kassel and is now a Professor for control theory and automation engineering at Jade University of Applied Sciences. Her main research interests are computational optimisation methods and intelligent algorithms for applications in control.

OLIVER ZIELINSKI is head of the research group "Marine Sensor Systems" at the Institute for Chemistry and Biology of the Marine Environment (ICBM), Carl von Ossietzky University of Oldenburg. He is also heading the research group "Marine Perception" at the German Research Center for Artifical Intelligence (DFKI). After receiving his Ph.D. degree in Physics in 1999 from University of Oldenburg, he moved to industry where he became scientific director and CEO of "Optimare Group," an international supplier of marine sensor systems. In 2005, he was appointed Professor at the University of Applied Science in Bremerhaven, Germany. In 2007, he became Director of the Institute for Marine Resources (IMARE). He returned to the Carl von Ossietzky University of Oldenburg in 2011. His area of research covers marine optics and marine physics, with a special focus on coastal systems, marine sensors, and operational observatories involving different user groups and stakeholders.