Proceedings of the 2019 Winter Simulation Conference
N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, eds.

SEGOLS: A LINE-BASED SEQUENTIAL SAMPLING STRATEGY FOR EFFICIENT DESIGN SPACE EXPLORATION AND OPTIMIZATION

Rémi Delanghe
Tom Van Steenkiste
Dirk Deschrijver
Tom Dhaene

IDLab
Ghent University - imec
Technologiepark-Zwijnaarde 126
Ghent 9052, BELGIUM

ABSTRACT

Sampling in a physical measurement space is omnipresent in both scientific studies and engineering applications. An important consideration is the length of the path between consecutively visited samples. A line-based sampling strategy both minimizes the path length and optimizes the sample locations, while maximizing the gain of information.

The state of the art on design of experiments envelopes a plethora of explorative line-based sampling techniques that solely focus on exploring the measurement space. None of these techniques consider the measured sample values. However, an exploitation of these values offers the possibility to focus on interesting regions during the sampling process.

In this paper, an exploitative line-based sampling technique, called SEGOLS, is proposed that balances exploration of the measurement space and exploitation of the interesting regions. The algorithm is illustrated using a 2D benchmark and an engineering measurement use-case. The results show that SEGOLS offers an improvement over explorative sampling techniques.

1 INTRODUCTION

In many fields of science and engineering, measurements are an essential way to acquire a representation of an underlying, often complex, function. Determining a sampling strategy is often done through Design of Experiment (DoE) methods (Kleijnen 2015). These methods aim at optimizing the locations of the samples such that the information gain per sample is maximized. There is a clear trade-off between the amount of samples taken and the associated cost of the experiment. Increasing the amount of samples can increase the quality of the representation, but will also increase the cost of the complete set of experiments.

When measurements are taken in a virtual space, such as a parameter space in a simulation, there is no constraint on the locations of sequentially taken samples. In a physical measurement space, using a measurement probe, the probe has to physically move and hence all samples form a path with a predetermined order. Examples include Electromagnetic compatibility (EMC) testing (Deschrijver et al. 2012), organ palpation (Salman et al. 2018) and vegetation monitoring (Berni et al. 2009). The length of this path has to be taken into account in order to minimize sampling cost, as moving the probe takes measurement time, possibly resulting in an economic cost. The path length might also be limited due to physical constraints, for instance battery lifetime or the chance of mechanical failure (Carlson and Murphy 2005). However, as sequentially taken samples are close together, the additional information of a subsequent sample is not optimal. Here, a new trade-off emerges. On the one hand, the space between samples should be minimized to maximize the amount of samples on the path, and on the other hand, the space should be sampled as uniformly as possible.

For some applications, it might be interesting not only to capture the underlying function, but to focus on particular regions of interest. The measurement probe can explore the complete space, or it can search for specific information in the space by exploiting the already measured data. This might be a section of the measurement space where the function has a maximum value or the location where the function exceeds a certain threshold. In EMC tests for instance, being able to quickly identify if a threshold is exceeded, is more important than gathering a detailed representation of the electromagnetic field. To achieve this, the sampling algorithm should contain an exploitative element which uses previous sample values to estimate where interesting regions in the measurement space might reside. This involves a trade-off between the exploration of the measurement space which allows to discover new regions of interest and exploitation which allows for more detailed representation of the measurement space in these regions.

The state of the art in line-based sequential DoE is currently limited to explorative sampling strategies (Van Steenkiste et al. 2019). In this paper, the Sequential Exploitative Global-Optimization Local-Steering (SEGOLS) sampling strategy is presented. It uses a Gaussian Process surrogate model and an acquisition function to sequentially trace out a path of sampling locations through the measurement space. Given an adequate acquisition function for the sampling problem at hand, the algorithm balances between exploring the complete region and exploiting the captured data to focus the sampling on finding the maximum value of the measurement space.

The paper is organized as follows. Section 2 will go deeper into the current state of the art, presenting the different properties by which DoE methods can be characterized. In Section 3, the use of a Gaussian Process surrogate model to represent the captured data is explained. Next, in Section 4, the novel SEGOLS sampling strategy, is presented. Section 5 presents the experimental setup and the results. Finally in section 6 conclusions are drawn.

2 CHARACTERISTICS OF DOE METHODS

All DoE methods have the common goal of defining a sampling strategy in a measurement space. These DoE methods can be characterized using several properties.

2.1 Point Based versus Line-based

A DoE method is classified as point based when it does not take into account the cost of movement between sample points. This is, for example, the case for sampling methods within a virtual measurement space such as in computer simulations. Some examples include factorial designs (Montgomery 2017) and FLOLA-voronoi sampling (Van Der Herten et al. 2014).

In line-based sampling, the cost of moving around in the measurement space is taken into account. All sample points are taken in a specific order and form a path. The information gain per sample should be maximized while keeping the path length to a minimum. Line-based sampling is very similar to coverage path planning (Galceran and Carreras 2013), where the measurement probe must pass through every point in the measurement space. Examples include the Boustrophedon path (Choset and Pignon 1998) and the Hilbert curve (Sadat, Wawerla, and Vaughan 2015).

2.2 One-Shot versus Sequential

When a fixed number of sample points N is determined at the start of the DoE method, it is characterised as a one shot method. An example is factorial designs (Montgomery 2017). Determining the amount of samples upfront, allows for an ideal placement and coverage of the space. However, determining the optimal amount of samples N can be difficult and underestimations or overestimations can lead to additional costs.

Table 1: Classification of sample strategy examples. The distinction is made between point-based and line-based, between one-shot and sequential and between purely explorative and containing exploitative elements. The SEGOLS sampling strategy fills in the gap of sequential exploitative line-based sampling strategies.

Examples	**Point-based VS Line-based**	**One-shot VS Sequential**	**Uses Exploitation**
Factorial Designs (Montgomery 2017)	Point-based	One-shot	No
Boustrophedon path (Choset and Pignon 1998) Hilbert curve (Sadat, Wawerla, and Vaughan 2015)	Line-based	One-shot	No
Maximin criterion (Johnson et al. 1990)	Point-based	Sequential	No
ALBATROS (Van Steenkiste et al. 2019).	Line-based	Sequential	No
-	Point-based	One-shot	Yes
-	Line-based	One-shot	Yes
EGO (Jones et al. 1998) FLOLA-voronoi (Van Der Herten et al. 2014)	Point-based	Sequential	Yes
SEGOLS (This work)	Line-based	Sequential	Yes

Sequential methods on the other hand, iteratively increase the amount of samples by determining consecutive sample locations based on the location and information gained from the previous samples. Theoretically, these algorithms could continue indefinitely. In practice, stopping criteria are used to determine when sufficient samples are available. Examples of sequential DoE include the point based maximin sampling (Johnson et al. 1990) and the line-based Albatros algorithm (Van Steenkiste et al. 2019).

The iterative process makes sequential methods more desirable in situations where it is uncertain how many samples will be needed before a certain criteria is reached or where mechanical failure of the measurement probe is likely.

2.3 Exploration versus Exploitation

A DoE method can be classified as explorative, when the measurement values are not taken into account to determine subsequent sample locations. These methods solely focus on space-fillingness to uniformly distribute samples within the measurement space. Examples include the Hilbert curve (Sadat, Wawerla, and Vaughan 2015) and the Albatros algorithm (Van Steenkiste et al. 2019).

This is in contrast to exploitative DoE methods that use the measured sample values in order to optimize the location of upcoming samples and to focus on a region of interest. This can for example be the maximum of the sampled function. It is evident that exploitative methods are sequential by nature. This is because the previous sample values have to be taken into account before new samples can be taken. An example of a point-based sequential technique is the efficient global optimization (EGO) algorithm (Jones et al. 1998) using bayesian optimisation (BO).

It is important to note that the distinction between explorative and exploitative methods is not binary. An explorative algorithm should encompass a degree of exploration in order to increase the knowledge about the environment which then can then be exploited. This trade-off between exploration and exploitation is a more general problem in the field of machine learning and artificial intelligence (Thrun 1995), (Auer 2002) and (Osugi et al. 2005) and in other fields like organisational management (Raisch et al. 2009).

Up until now, no exploitative line-based DoE method exists. This paper aims at filling in this gap as demonstrated in Table 1.

3 GAUSSIAN PROCESS SAMPLING MODEL

The use of machine learning and in particular Gaussian Processes (GP) in DoE is well-established. One such an example is the EGO algorithm (Jones et al. 1998). The use of a non-parametric GP model to estimate the values of the complete measurement space using few data-points makes it a desirable technique which does not only return an estimation of the measurement space (GP Mean) but also a degree of uncertainty (GP Variance) on this estimation. In the EGO algorithm, the GP Mean, GP Variance and the sample values are used to calculate the best next sample location in the complete measurement space. The best sample location is the location which has the most potential to reveal the maximum (or minimum) of the measurement space. This is done by using a so-called acquisition function such as the Expected Improvement (EI), Probability of Improvement (PoI) or the Upper Confidence Bound (UCB) (Snoek, Larochelle, and Adams 2012). The acquisition function is calculated using the GP Mean and GP Variance. The function outputs a desirability score for each point in the measurement space. In a point-based bayesian optimization the acquisition funtion is maximized which reveals the most desirable locations to sample at. The acquisition function offers an automatic trade-off between exploration (sampling at locations with high uncertainty) and exploitation (sampling at locations with high) (Brochu, Cora, and De Freitas 2010).

In this work, the UCB acquisition function is considered which is given by (1) where, μ represents the GP mean, σ represents the GP variance. For every location of the measurement space the acquisition function takes the weighted sum of the GP mean and GP variance at that location.

$$\mathrm{UCB} = \mu(x) + k\sigma(x) \tag{1}$$

When little samples are taken, the overall uncertainty will be high and the second term of the equation will dominate. This will result in more explorative behaviour. When more samples are taken, the influence of the first term will increase and with it, the behaviour will become more exploitative. The parameter k influences the point in time at which the exploitative behaviour starts overtaking the explorative one. When k is small, the contribution of the GP Mean will quickly outweigh that of the GP variance. However, when k is large, more samples have to be taken before the GP variance is reduced enough for the GP Mean to dominate the acquisition function. As in any of the above mentioned parametric acquisition functions, UCB introduces the parameter k which is left to the user (Brochu, Cora, and De Freitas 2010).

4 SEQUENTIAL EXPLOITATIVE GLOBAL OPTIMIZATION LOCAL STEERING SAMPLING STRATEGY (SEGOLS)

The Sequential Exploitative Global-Optimization Local-Steering sampling strategy (SEGOLS) is an exploitative sequential line-based strategy. It can be considered as an extension of the traditional BO sampling strategy with the added constraint of line-based movements.

The algorithm starts by taking an initial sample, taken at an arbitrary location. Next the first iteration starts by modelling the gathered data with the GP and computing the maximum of the acquisition function. This point is the waypoint for the next path extension. This is the Global Optimization (GO) step of the algorithm. Maximizing the acquisition function is not a trivial task. For a small number of dimensions, the maximisation can be performed by uniformly sampling the space. For higher dimensions, a more advanced technique like proposed in (Wilson, Hutter, and Deisenroth 2018) can be employed. The path towards the maximum is calculated by iterating over the following logic which is referred to as a subiteration:

- If the point is within reach, meaning that the location of the measurement probe is closer than a single step-size to the location of the waypoint: the probe will move to the location and take a sample.
- Else, the best point within view is chosen. This is achieved using an arc that defines the view-angle, centered at the probe location and oriented as such that the line between the probe and the waypoint cuts the arc in two arcs of the same angle. Along this arc, the acquisition function is evaluated and

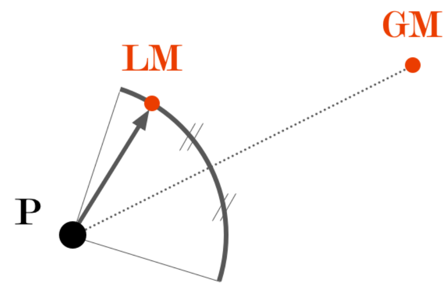


Figure 1: Illustration of the local steering aspect of the algorithm. With P, the sampling probe; LM, the local maximum on the view arc; GM, the Global maximum. Both the step-size, represented by the arrow, and the view angle are user-specified.

the probe moves and samples at the maximal evaluated point. This is illustrated in Figure 1. This is the Local Steering (LS) step of the algorithm which involves local manoeuvring of the probe based on the values of the acquisition function in the neighborhood of the probe and the user-specified values of the view-angle and the step-size.

This subiteration is repeated until the waypoint is reached. After this, a new waypoint is defined given the new samples and a new iteration is started. If the view-angle is smaller than π radians, the waypoint is guaranteed to be reached and bounds for the number of steps can be calculated. As in point-based BO algorithms, the sampling can go on indefinitely. However, the algorithm is stopped when 2 consecutive waypoints are picked at the same location.

This is referred to as the Greedy case. The pseudo code of the algorithm can be found in Algorithm 1. Note that the GP is not refit before the waypoint is reached, for two reasons:

- When the probe comes closer to the waypoint, the information to be gained by sampling the waypoint itself reduces. This will make the algorithm turn away from potentially interesting sample locations.
- Fitting the GP for many points in every subiteration, is very computationally expensive when using a classical implementation of GP (Herbrich et al. 2003). By including this calculation only at the beginning of an iteration, calculating the path is much more feasible which makes the algorithm useful in real time applications.

As mentioned in Section 4, the acquisition function itself makes the trade-off between exploration and exploitation. By using the UCB acquisition function, SEGOLS starts by exploring the space and as the uncertainty of the GP model reduces, the behaviour will become more exploitative.

5 EXPERIMENTAL SETUP

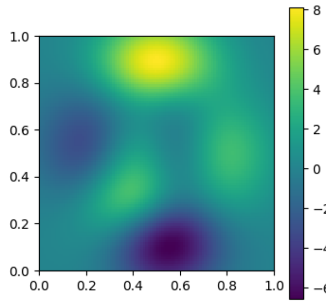
5.1 Setup

The SEGOLS Sampling Strategy will be compared against other DoE techniques mentioned above. The algorithms are tested on the peaks benchmark function, given by (2).

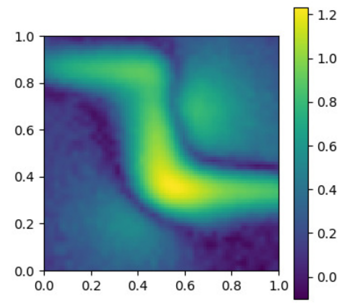
```
Data:
  • initial_sample   // Sample of the starting position of the probe
  • model            // The Gaussian Process with chosen acquisition function
  • probe            // Measurement probe containing location of the probe
  • view_angle       // Parameter to tune the view angle, used in the sub iteration
  • step_size        // Parameter to tune the maximum step size between consecutive samples.

// initialization
all_samples = initial_sample;
model.train(all_samples);
waypoint = ∅;
// Start of the algorithm
while end_condition_not_met do
    // Set a new waypoint at the global maximum of the acquisition function
    new_waypoint = model.acquisition_function.get_maximum_location();
    // Check if the algorithm still finds a new global maximum
    if new_waypoint is waypoint then
        break;
    else
        waypoint = new_waypoint;
        // Start of a sub iteration
        while distance(probe.position, waypoint) > step_size do
            // Get the maximum position of the acquisition function on the view
            sub_waypoint = model.acquisition_function. \
                           get_maximum_location_on_arc(view_angle, probe.position, waypoint);
            // Sample at this position
            probe.position = sub_waypoint;
            new_sample = probe.sample_at_current_location();
            all_samples.append(new_sample);
        end
        // The current calculated global maximum of the acquisition function is in reach
        // Sample at this point
        probe.position = waypoint;
        new_sample = probe.sample_at_current_location();
        all_samples.append(new_sample);
    end
end
```

Algorithm 1: Overview of the SEGOLS Sampling Strategy.



(a) Peaks function.



(b) EMC measurement data.

Figure 2: Representations of the measurement data

$$
\begin{aligned}
f(x,y) = & 3(-1-4x)^2 \times exp(-((4x-2)^2) - (4y-1)^2) \\
& - 10((1/5) \times (4x-2) - (4x-2)^3 - (4y-2)^5) \times exp(-(4x-2)^2 \\
& - (4y-2)^2) - (1/3) \times exp(-(4x-1)^2 - (4y-2)^2)
\end{aligned}
\tag{2}
$$

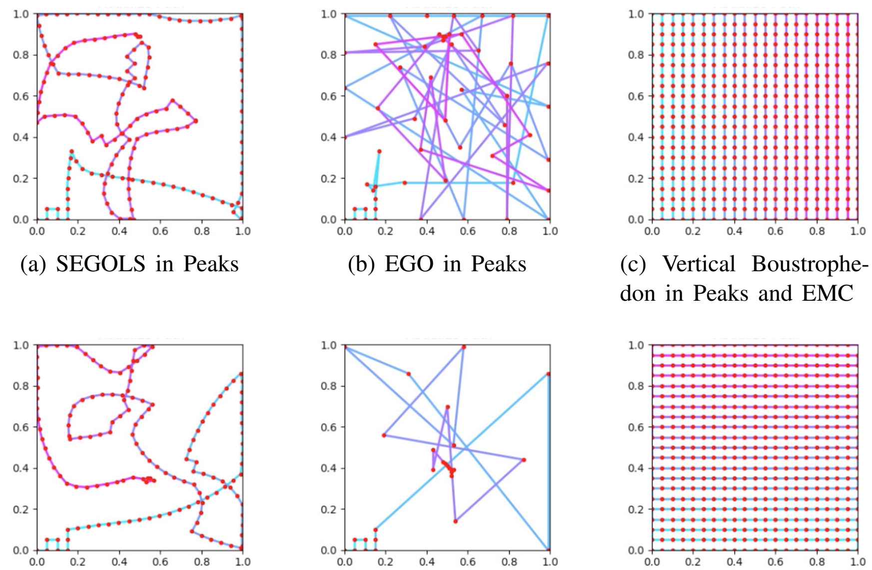
A 2D representation of the function can be seen in Figure 2(a). The algorithm is also tested on an engineering problem, namely an electromagnetic compatibility (EMC) testing use-case. A near field scanning probe scans a device under test in order to capture the electric or magnetic field in the plane. A more detailed description of the setup can be found in (Deschrijver et al. 2012). In this paper, the authors collected a one shot dense grid scan which uniformly samples the magnetic field in the x plane (H_x) of a bent microstrip resulting in 3375 sample points. A subset of the data, focused on the most interesting region of the use-case, is taken in combination with bilinear interpolation in order to represent the unknown function in the experiment. The resulting representation is shown in Figure 2(b).

The tested algorithms are the SEGOLS Sampling Strategy with view-angle $\pi/4$ radians, the Boustrophedon path and an adapted version of the EGO algorithm. The EGO algorithm was altered such that it does not start with an initial set of uniformly located samples and such that no diagnostic tests are run after each iteration as is the case in the original implementation. This was done such that all strategies are subjected to the same restrictions. The EGO algorithm and the SEGOLS sampling strategy, use the UCB acquisition function where k has been given the value 10. The exploitative strategies are run eight times in both measurement spaces until completion, starting twice from each corner with a different set of eight initial samples. In the case of the Boustrophedon path, the horizontal and vertical variant of the algorithm are performed in each corner, also resulting in eight runs. The line based techniques, are tested with a step-size a twentieth of the measurement space (step-size = 0.05). To calculate the path length of the EGO algorithm, the probe is considered to move straight between sample points.

5.2 Evaluation

In order to evaluate the performance of the different algorithms, two criteria are evaluated.

Firstly, the similarity between the maximum of the collected data samples and the maximum of the ground truth is evaluated. This measures the ability of the strategy to be exploitative and find the maximum of the sampled space. To achieve this, the sample with the highest value of all already collected samples is compared to the maximum of the ground truth as illustrated by the Maximum Sample Value (MSV) (3). When comparing multiple strategies, the strategy which converges the fastest to value 1 would be considered the better strategy.



(a) SEGOLS in Peaks (b) EGO in Peaks (c) Vertical Boustrophedon in Peaks and EMC

(d) SEGOLS in EMC (e) EGO in EMC (f) Horizontal Boustrophedon in Peaks and EMC

Figure 3: Examples of samples trajectories taken by the different sampling strategies. The red dots represent the samples. The trajectory is colored using a gradient from blue to purple respectively indicating the start and ending of the path.

$$\text{Maximum Sample Value} = \left| \frac{\max_i(f(sample_i))}{\max_{x,y}(f(x,y))} \right| \tag{3}$$

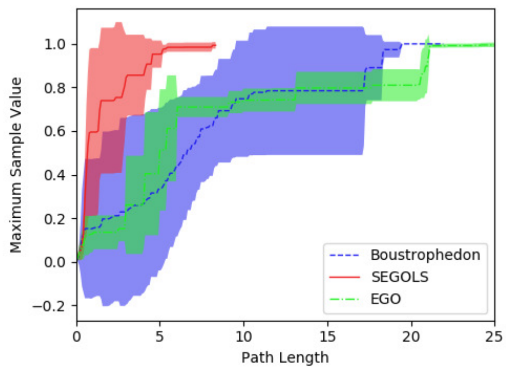
Secondly, the error between the estimated function and the ground truth is calculated. Both the GP mean of the estimated function and the original ground truth are uniformly sampled using 10,000 samples. The predicted sample values $pred(x,y)$ of the GP mean are compared with the samples of the ground truth $f(x,y)$ using the RMSE metric (4). This metric evaluates the ability of the algorithms to capture the measurement space.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(pred(x_i,y_i) - f(x_i,y_i))^2} \tag{4}$$

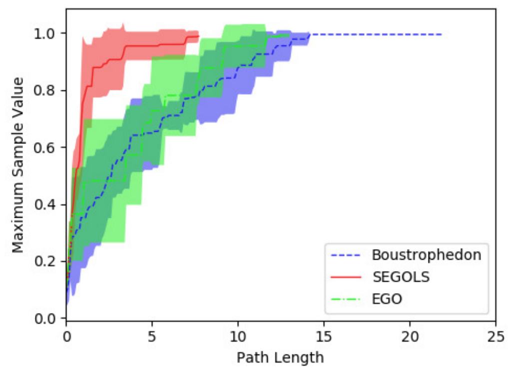
These metrics are evaluated with respect to the path length of the data collection strategies. In order to validate the difference between the SEGOLS strategy and the other strategies with respect to these metrics, the Kruskal-Wallis H-test (KWH-test) was employed. The obtained values over the multiple runs for a certain path length of the SEGOLS strategy are pairwise compared to the values of the other strategies. The null hypothesis is that there is no significant difference between the measured values for acertain path length. If the $p-value$ for this path length is smaller than 0.05, the null hypothesis is rejected and the difference is considered to be significant.

5.3 Results

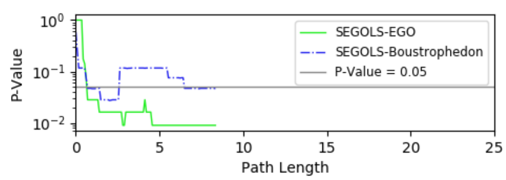
Example trajectories from the execution of the strategies are shown in Figure 3. The results of the experiments can be found in Figure 4. Figure 4 shows aggregated information for the runs. At every value of the path length, the average measured value of the metric is depicted together with the standard deviation. All runs are included in Figure 4 except for 3 runs from the EGO EMC experiments. In these



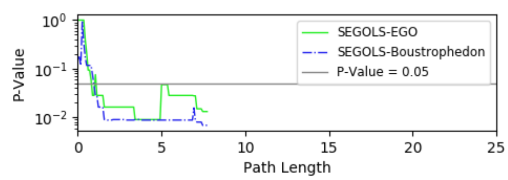
(a) MSV per Path Length (Peaks)



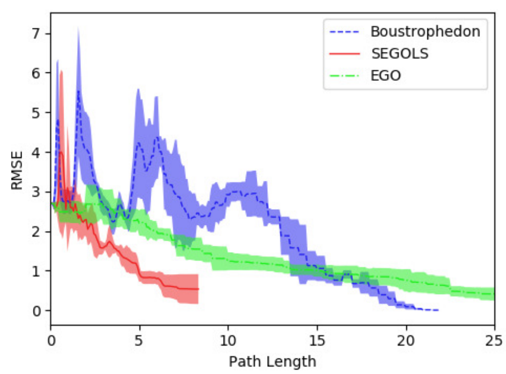
(b) MSV per Path Length (EMC)



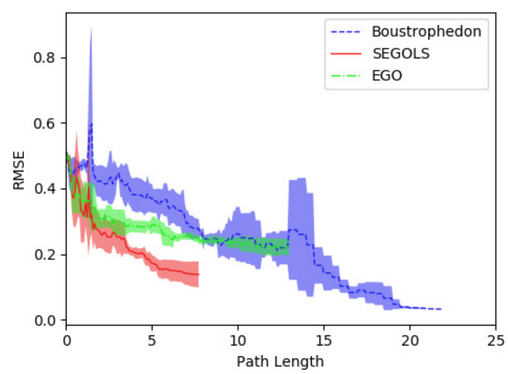
(c) P-Value of KWH-test for MSV per Path Length (Peaks)



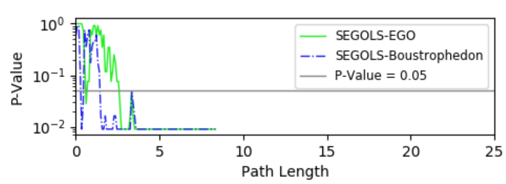
(d) P-Value of KWH-test for MSV per Path Length (EMC)



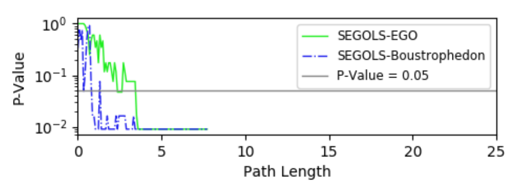
(e) RMSE per Path Length (Peaks)



(f) RMSE per Path Length (EMC)



(g) P-Value of KWH-test for RMSE per Path Length (Peaks)



(h) P-Value of KWH-test for RMSE per Path Length (EMC)

Figure 4: Evolution of the MSV and RMSE of the Peaks and EMC experiments for the SEGOLS, EGO and Boustrophedon strategies in function of path length. Every evolution of a metric is accompanied by the evolution of the P-Value resulting from the pairwise of KWH-test.

cases the strategy got stuck in a local maximum and are considered outliers. The evolution of every metric is accompanied by the evolution of the P-Value resulting from the pairwise of KWH-test. The gray horizontal line indicates threshold beneath which the null hypothesis is rejected and the difference between the strategies are significant.

A first observation is that the SEGOLS experiments, on average, performs better than the other strategies for both metrics. Indeed Figure 4(a) and Figure 4(b) show that SEGOLS approaches the maximum of the spaces the fastest and 4(e) and Figure 4(f) show that the RMSE is reduced most rapidly using the SEGOLS strategy. The EGO and Boustrophedon path are comparable in performance on both metrics. By observing Figures 4(c), 4(d), 4(g), 4(h) it is clear that the difference in performance is significant. The only exception is for the MSV in the Peaks experiment where the null hypothesis is not rejected in the comparison between SEGOLS and the Boustrophedon path. This is due to the high variance on the performance of the Boustrophedon path for this metric. Indeed, the optimal location of the peaks function lies at $(0.49, 0.89)$. The different runs of the Boustrophedon paths, which purely explore the space in an uninformed manner, will discover the maximum at very different times, leading to the high variability. This is less prevelant in the EMC experiments because the maximum lies closer to the middle of the space $(0.53, 0.35)$ and will be discovered roughly at the same time.

Even though the EGO algorithm samples at the best new sample location according to the acquisition function, the MSV and RMSE metrics respectively increase and decrease slowly. This is because consecutive samples may lay far apart from each other and there is no way for the EGO algorithm to increase its performance while traveling from one optimal sample location to another. Thus, even though the EGO algorithm might outperform the other strategies for these metrics in function of sample count, it is outperformed by the SEGOLS strategy because it does sample on its path towards the ideal sample location. The Boustrophedon path suffers from a different problem. Looking at Figure 3(c) and Figure 3(f) shows that the Boustrophedon always travels in areas which are already densely sampled. This means that very little information is gained over time.

A final observation with regards to Figure 4 is that the SEGOLS strategies finish with a relative short path, compared to the other strategies. This is because, once the Acquisition function gets stuck in an optima, no new waypoints are found and the SEGOLS strategy does not progress anymore. However, the Boustrophedon paths do continue for longer, which allows them to keep decreasing their RMSE score and finish with a lower RMSE score. This means that if accuracy is more important than speed or path length, the one-shot Boustrophedon path outperforms SEGOLS.

5.4 Time Complexity

Evaluating the complexity of SEGOLS analytically is a non-trivial task. The complexity depends among other on the technique used to fit the data into the Gaussian Process, the technique used to optimize the model and the technique used to maximize the acquisition function. In a physical experiment, the speed of movement of the measurement probe will also influence the time needed to execute the experiment.

A more feasible task is to compare execution time between the different strategies. In this analysis, the travel time between consecutive samples is not taken into account because of the reason mentioned above. Table 2 shows the average execution time per path length and per number of samples of the EGO algorithm and Boustrophedon path, with respect to the execution time of the SEGOLS sampling strategy. This was measured using the experiments from the previous section. The Boustrophedon path has a very low execution time, being multiple orders of magnitude faster than the SEGOLS strategy. This is because no model has to be calculated at every iteration. When using the same configuration of the model, execution time of the EGO strategy with respect to the path length is similar to the execution time of the SEGOLS strategy, but when considering the execution time per number of samples, the EGO algorithm is 6.5 times slower. The reason being that, even though consecutive waypoints in the SEGOLS strategy are approximatly at the same distance as consecutive samples in the EGO algorithm, the SEGOLS strategy also samples

Table 2: Table showing the relative execution time per unit of traveled path and the relative execution time per captured sample of the SEGOLS, EGO and Boustrophedon strategies. In both cases, the execution time is compared to the execution time of the SEGOLS sampling strategy. The actual average execution time for the SEGSOLS strategy was on average 168 ms per sample and 3790 ms per unit of length.

Algorithm	**SEGOLS**	**EGO**	**Boustrophedon**
Time / Path Length	1.000	1.007	3.430×10^{-4}
Time / Number of Samples	1.000	6,586	3.9715×10^{-4}

along the path towards the waypoint. As such SEGOLS samples approximatly 6.5 times more samples per iteration.

6 CONCLUSION

In this paper, the SEGOLS Sampling strategy was introduced, benchmarked and applied to an existing engineering problem. The aim of the algorithm is to fill in the gap in DoE algorithms with a line-based, sequential and exploitative algorithm. The algorithm uses a Gaussian Process as underlying model to define waypoints between which to travel and to construct a path between the waypoints. The use of the Gaussian Process allows for balancing exploration and exploitation within the algorithm.

Whereas point based strategies optimize a metric in terms of the number of samples, line-based strategies focus on spreading the amount of samples while optimizing the path length. The SEGOLS sampling strategy augments the capabilities of line based sample strategies by dynamically focusing on regions of interest and on regions with limited knowledge. For both the benchmark experiment and the EMC engineering use-case the use of SEGOLS shows a significant improvement to capture the underlying function and discover the global maximum while decreasing the total path length. This both increases the speed and decreases the cost of the sampling process which makes the SEGOLS sampling strategy especially interesting for time consuming experiments as well as in situations where the amount of samples is not known upfront or where failure of the measurement probe is possible.

REFERENCES

Auer, P. 2002. “Using confidence bounds for exploitation-exploration trade-offs”. *Journal of Machine Learning Research* 3(Nov):397–422.

Berni, J. A., P. J. Zarco-Tejada, L. Suárez, and E. Fereres. 2009. “Thermal and narrowband multispectral remote sensing for vegetation monitoring from an unmanned aerial vehicle”. *IEEE Transactions on Geoscience and Remote Sensing* 47(3):722–738.

Brochu, E., V. M. Cora, and N. De Freitas. 2010. “A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning”. *arXiv preprint arXiv:1012.2599*.

Carlson, J., and R. R. Murphy. 2005. “How UGVs physically fail in the field”. *IEEE Transactions on robotics* 21(3):423–437.

Choset, H., and P. Pignon. 1998. “Coverage path planning: The boustrophedon cellular decomposition”. In *Field and service robotics*, 203–209. Springer.

Deschrijver, D., F. Vanhee, D. Pissoort, and T. Dhaene. 2012. “Automated near-field scanning algorithm for the EMC analysis of electronic devices”. *IEEE Transactions on Electromagnetic Compatibility* 54(3):502–510.

Galceran, E., and M. Carreras. 2013. “A survey on coverage path planning for robotics”. *Robotics and Autonomous systems* 61(12):1258–1276.

Herbrich, R., N. D. Lawrence, and M. Seeger. 2003. “Fast sparse Gaussian process methods: The informative vector machine”. In *Advances in neural information processing systems*, 625–632.

Johnson, M. E., L. M. Moore, and D. Ylvisaker. 1990. “Minimax and maximin distance designs”. *Journal of statistical planning and inference* 26(2):131–148.

Jones, D. R., M. Schonlau, and W. J. Welch. 1998. “Efficient global optimization of expensive black-box functions”. *Journal of Global optimization* 13(4):455–492.

Kleijnen, J. P. 2015. “Design and analysis of simulation experiments”. In *International Workshop on Simulation*, 3–22. Springer.

Montgomery, D. C. 2017. *Design and analysis of experiments*. John wiley & sons.

Osugi, T., D. Kim, and S. Scott. 2005. "Balancing exploration and exploitation: A new algorithm for active machine learning". In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, 8–pp. IEEE.

Raisch, S., J. Birkinshaw, G. Probst, and M. L. Tushman. 2009. "Organizational ambidexterity: Balancing exploitation and exploration for sustained performance". *Organization science* 20(4):685–695.

Sadat, S. A., J. Wawerla, and R. Vaughan. 2015. "Fractal trajectories for online non-uniform aerial coverage". In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2971–2976. IEEE.

Salman, H., E. Ayvali, R. A. Srivatsan, Y. Ma, N. Zevallos, R. Yasin, L. Wang, N. Simaan, and H. Choset. 2018. "Trajectory-optimized sensing for active search of tissue abnormalities in robotic surgery". In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 1–5. IEEE.

Snoek, J., H. Larochelle, and R. P. Adams. 2012. "Practical bayesian optimization of machine learning algorithms". In *Advances in neural information processing systems*, 2951–2959.

Thrun, S. 1995. "Exploration in active learning". *Handbook of Brain Science and Neural Networks*:381–384.

Van Der Herten, J., D. Deschrijver, and T. Dhaene. 2014. "Fuzzy local linear approximation-based sequential design". In *2014 IEEE Symposium on Computational Intelligence for Engineering Solutions (CIES)*, 17–21. IEEE.

Van Steenkiste, T., J. van der Herten, D. Deschrijver, and T. Dhaene. 2019, Apr. "ALBATROS: adaptive line-based sampling trajectories for sequential measurements". *Engineering with Computers* 35(2):537–550.

Wilson, J., F. Hutter, and M. Deisenroth. 2018. "Maximizing acquisition functions for Bayesian optimization". In *Advances in Neural Information Processing Systems*, 9884–9895.

AUTHOR BIOGRAPHIES

RÉMI DELANGHE is a master student at Ghent university. The subject of his master thesis is Model-driven approaches for sequential line-based data collection. In the context of his thesis, he works on sequential line based sampling techniques. His email address is remi.delanghe@ugent.be.

TOM VAN STEENKISTE received his M.Sc. degree in Computer Science Engineering from Ghent University in 2016. Since August 2016, he is active as a PhD student in the Internet and Datascience Lab (IDLab) research group at Ghent University where he is working on data analysis techniques for medical time series data and sensitivity analysis for surrogate modeling. His current research interests are eHealth, time series analysis, surrogate-based Optimization and machine learning in general. His email address is tomd.vansteenkiste@ugent.be.

DIRK DESCHRIJVER is as an associate professor in the IDLab research group of Ghent University, working on data analytics, machine learning and surrogate modeling algorithms. He received the Master degree in Computer Science in 2003, from the University of Antwerp, Belgium. Since then, he worked as a PhD student in the Dept. of Mathematics and Computer Science at the same university, where he obtained the PhD degree in 2007. During the period of May-October 2005, he was a Marie Curie Fellow in the Scientific Computing group at the Eindhoven University of Technology in Eindhoven, The Netherlands. In 2006 and 2008, he was also a visiting researcher at SINTEF Energy Research in Trondheim, Norway and the University of L'Aquila in Italy. His research was mainly focused on the macromodeling of high-speed and high-frequency interconnects, signal integrity and electromagnetic compatibility. From 2008-2014, he worked as an FWO post-doctoral research fellow in the IBCN research group in the Department of Information Technology at Ghent University. In 2012, he obtained a second PhD degree degree, in engineering, at Ghent University. Since October 2014, he has been working as a senior researcher at imec. His email address is Dirk.Deschrijver@UGent.be.

TOM DHAENE is a professor at the IDLab research group at Ghent university - imec for *Machine Learning* and *Algorithms and Data structures* in the engineering department. Some current research focuses are Experimental Design, Computer-aided Design, Supervised Machine Learning, Sequential sampling and Computational Electromagnetics. Tom Dhaene received the M.S. degree and the Ph.D. degree in Electrical Engineering from Ghent University, Ghent, Belgium, in 1989 and 1993, respectively. From 1989 to 1993, he was Research Assistant at Ghent University, in the Department of Information Technology (INTEC), where his research focused on different aspects of full-wave electromagnetic circuit modeling, transient simulation, and time-domain characterization of high-frequency and high-speed interconnections. His e-mail address is Tom.Dhaene@UGent.be.