



**Országos Tudományos Diákköri
Konferencia
2025**



**Analysis and Clustering-based Improvement
of Particle Filter Optimization Algorithms**

Készítette: Kenyeres Éva

Témavezető: Prof. Abonyi János

Pannon Egyetem, Mérnöki Kar, Folyamatmérnöki Intézeti Tanszék

A pályamunka lezárásának dátuma: 2023.10.31.

Contents

1	Introduction	2
2	Tuning of PFO for global search and exploration of local optimums	5
2.1	The general particle filter optimization (PFO) algorithm	5
2.2	Interpretation of the elements and data-driven improvements of the general particle filter optimization (PFO) algorithm	9
2.2.1	Distribution of the observation noise	9
2.2.2	Proposals to ensure the exploration of the search space	11
2.2.3	Stopping criterion issues	12
2.3	A clustering-based method for exploring the shape of the objective function	13
2.4	The extended form of the PFO algorithm	14
3	Optimisation experiments on the Michalewicz-function	16
3.1	Analysis and improvements of the efficiency of the PFO algorithm for global search	17
3.2	Clustering-based technique for finding local optimums	20
3.3	Factors that affect the local optimum search performance	27
4	Conclusions	33
A	My Appendix	35
A.1	PF framework	35
A.2	Systematic resampling	36

Abstract

This study highlights how particle filter optimization (PFO) algorithms can explore objective functions and their robustness near to optimums. Improvements of the general algorithm are also introduced to increase the efficiency of the search.

Population-based optimization algorithms gained more interest in the last few decades. They reach better exploration of the search space, thereby, higher efficiency by propagating not only one but many candidate solutions which can also be stored. One novel representative of these methods is the PFO concept. Similarly to its counterpart for state estimation, the algorithm results a probability distribution of the sample elements, which can represent the shape of the objective function in this case. In the literature, several variants of the PFO can be found and its elements are not clearly fixed because of its novelty. Its research moves on the theoretical level yet, mainly for finding the global optimum, however, there was not find a work which investigates the additional information can be extracted by the analysis of the distribution of the sample set.

In the present study, a method is introduced for gaining more information about the shape of the objective function by following the propagation of the particles along the iterations. Besides, a comparative analysis is proposed examining the different variants of the algorithm, and some improvements are also introduced to increase the efficiency of the search. The contributions of the work: 1) A detailed description of the PFO algorithm and considerations on its elements are provided; 2) A comparative study is introduced in order to improve the efficiency of the general algorithm; 3) A clustering-based technique is proposed to explore the shape of the objective function; 4) The propagation of the particles is investigated to get information about the complexity of the objective function and its local optimums. The experiments are executed on the Michalewicz function owning sharp peaks and many local optimums thus defining a difficult search which is also configurable by some parameters. The results verify that with the proposed method, the complexity and shape of the objective function can be explored which is a promising prospect for robust optimization problems where often not the global, but a more stable local optimum gives best the solution.

Keywords: optimization of complex systems, clustering, particle filter, population-based optimization

Chapter 1

Introduction

The subject of this work is to investigate the improvement possibilities of a novel stochastic optimization algorithm called particle filter optimization (PFO) and extend the global search algorithm to be able to explore the objective function and gain information about its complexity.

Population-based optimization algorithms have gained wide interest in the last two decades. The increased speed of computations made it possible to use these algorithms efficiently, in contrast to the previous ages. As they store the whole group of candidate solutions, their result contains more information than that of the single solution algorithms [1], which may be worth exploring to gain as much information from the population as possible. As another advantage, the user does not need to define the starting point of the search, as the candidate solutions are placed randomly at the beginning of the search covering the whole space of the decision variables. One main representative of these algorithms is the particle filter optimization (PFO) method.

Particle filters (PF) were originally developed for state estimation, filling a niche in the case of nonlinear and noisy systems [2]. The fundamental idea behind the PF is to represent the states of the dynamic system by a discrete sample set. The sample elements are called particles and weighted according to some measurement data, thus defining not only the expected value of the states but also their probability distributions [3].

The probabilistic approach of the PFs was transplanted to the optimization field by converting the optimization problem to a filtering one [4]. In this case, the weights of the particles are determined according to the objective function value belonging to the candidate solutions during each iteration. Similarly to the PFs, the possibility to integrate uncertainties into the algorithm makes PFO as a sampling-based technique promising for robust optimization problems [5]. A multiobjective optimization variant of PFO has also been introduced in the literature recently [6].

Currently, the research on the PFO technique is at a theoretical level yet. Consequently,

some elements of the algorithm are flexible and some elements are not clearly defined. Therefore, it needs profound examination before application to choose the appropriate variant of the method. The algorithm has been tested mainly on benchmark functions before [7], and practical case studies are quite rare yet [8].

In order to make a robust optimization search more efficient, it is worth getting to know and understand better the shape of the objective function. The distribution of the sample set resulting from PFO can be well used for this aim, such as its propagation along the iterations. Clustering was combined with other population-based algorithms (e.g. GA in [9] and PSO in [10]) for finding peaks of multimodal functions, however, overiewing the relevant literature, there was not found a work dealing with this issue in the case of the PFO. This study investigates this idea, moreover, taking advantage of the probabilistic background of the PFO, the robustness of the optimums is also evaluated.

This work aims to explore which information can be obtained from the particle set and its moving along the iterations. The algorithm is examined from the perspective of robust optimization. In this kind of problems, usually not the global, but a more stable local optimum gives the best solution. My goal is to demonstrate how the PFO is applicable to find local optimums besides the global one in a single run, proposing a clustering-based method. This achievement can serve as the basis for future robust optimization research in the practical field.

The experiments were executed on the Michalewicz benchmark function which can be easily parameterized and was already used for the test of the PFO thus giving the chance of comparison [11]. The novelty of the work is a clustering-based technique to find the points that the particles concentrate around, i.e., the local optimums of the problem. Thereby, the shape and complexity of the objective function can be explored, and some information gained about the robustness of the peaks, as well.

The contributions of the study are the following:

- An extensive study about the different variants of the PFO elements.
- Improvement of the PFO algorithm by introducing a trajectory on the transition kernel and weight differentiation besides an alternative stopping criterion to reach better search efficiency.
- A technique for the investigation of the particle set to explore the shape of the objective function.
- A clustering-based method to find local optimums and evaluate their robustness.
- Recommendations on the ideal form of the algorithm besides different user aims.

- Ideas for future research based on the results how the computational demand can be reduced without losses on the performance.

The roadmap of the study is the following. Section 2 contains a detailed description of the used algorithms and techniques: the PFO algorithm is introduced in Section 2.1, its proposed improvements to increase the search efficiency is presented in Section 2.2, and the clustering-based method for analyse the particle sets is described in Section 2.3. Section 3 focuses on the experimental results and their explanation: Section 3.1 analyses the efficiency of the suggested improvements of the PFO algorithm, Section 3.2 reveals how the monitoring of the propagation of the particle set can be used to find local optimums and evaluate their robustness, and Section 3.3 examines which factors of the algorithm have an effect on the found local optimums. Last but not least, Section 4 summarizes the main conclusions of the work and gives some ideas about possible future research directions based on the results.

Chapter 2

Tuning of PFO for global search and exploration of local optimums

As PF is originally a state estimation algorithm, the optimization problem has to be transformed into a filtering one to be able to use the PFO. The population of the candidate solutions allows us to gain more information during and at the end of the optimization about the objective function and its local optimums than a single solution optimizer, which issues are investigated in this section.

In Section 2.1, the unique form of the optimization problem is discussed, and the general variant of PFO is introduced in detail under the scope of PF, and my improvements are presented in Section 2.2. The proposed clustering-based technique for investigating the distribution of the particles is introduced in Section 2.3.

2.1 The general particle filter optimization (PFO) algorithm

The sampling-based PFO technique for optimization is derived from the particle filter (PF) framework used originally for the state estimation of dynamic systems.

The general PF framework consists of three steps that are presented in more detail in Appendix A.1 [3]:

- the *prediction step*, which is responsible for moving the sample elements called particles forward one time step according to the state transition function of the state-space model;
- the *correction step*, where the particles get weights according to their consistency with the noisy observation available from measurement, using the measurement function to calculate probabilities;

- and the *resampling step*, which is necessary to avoid the degeneracy problem, thus ensuring the particle set to be effective enough [12].

In the case of the PFO, in contrast to the conventional way in the optimization field, the optimization problem is defined as a maximization task for practical reasons:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmax}} H(\mathbf{x}) \quad (2.1)$$

where $\mathbf{x}$ refers to the vector of the decision variables, $\mathbf{x}^*$ to its optimal value and $H(\cdot)$ to the objective function.

To be able to use the PF framework, the problem needs to be transformed into an iterative filtering problem described by a state-space model as [13]:

$$\mathbf{x}_k = \tilde{\mathbf{x}}_{k-1} \quad (2.2)$$

$$y_k = H(\mathbf{x}_k^*) - v_k \quad (2.3)$$

where k refers to the iteration number in the present context, $\mathbf{x}_k$ to the candidate solutions in iteration k generally and $\mathbf{x}_k^*$ to the best one of them. v_k represents the so-called observation noise which is sampled from a positive and strictly increasing continuous probability density function (pdf) which depends on the actual particle set as described in Section 2.2.1 represented by $\varphi_k(\cdot)$, and y_k denotes the noisy observation belonging to the best particle in the k th iteration, which forms the basis of weighting the candidate solutions (particles) and selecting the better ones.

Equation 2.2 represents the motion of the particles along the iterations that corresponds to the prediction step of the PF. Equation 2.3 produces the observation that is used to weight the particles in the correction step. The particles get weights according to the objective function value ($H(\mathbf{x}_k)$) belonging to them relative to y_k .

The so-called target pdf ($\pi_k(\mathbf{x}_k)$) is a continuous representation of the distribution of the weighted particles. In the present work, the most commonly used variant of π_k is applied (Equation 2.4) which is derived from the Bayes' rule and can be defined recursively as [13]:

$$\pi_k(\mathbf{x}_k) = \frac{\varphi_k(H(\mathbf{x}_k) - y_k)\pi_{k-1}(\mathbf{x}_k)}{\int \varphi_k(H(\mathbf{x}_k) - y_k)\pi_{k-1}(\mathbf{x}_k)d\mathbf{x}_k} \quad (2.4)$$

where π_k represents the target pdf of the k th iteration which definition is changing by the y_k value iteration by iteration.

The continuous $\pi_k(\mathbf{x}_k)$ pdf can be formed by the discrete distribution of the particle set. Thereby, the normalized weight of the i th particle can be calculated as:

$$\pi_k(\mathbf{x}_k^i) \approx w_k^i = \frac{\varphi_k(H(\mathbf{x}_k^i) - y_k)w_{k-1}^i}{\sum_{i=1}^{N_s} \varphi_k(H(\mathbf{x}_k^i) - y_k)w_{k-1}^i} \quad (2.5)$$

where N_s denotes the number of the particles and $\mathbf{x}_k^i$ refers to i the element of the population in iteration k .

In this case, the increasing differentiation of the objective function values is ensured by the monotonically increasing y_k . If $y_k < y_{k-1}$ occurs, y_k is set equal to y_{k-1} . Consequently, the interval of the performance of particles with nonzero weight becomes narrower as the iterations progress. As the nature of the $\varphi_k(\cdot)$ function does not change and it is bounded by the lowest and highest objective function value of the current iteration (as will be defined in detail in Section 2.2.1), it is like actually smaller and smaller region of the search space is explored, concentrated around the optimum. The statement that the target pdf defined in Equation 2.4 converges asymptotically was proven in [14].

The PFO algorithms, similarly to the PF framework, also contain a resampling step to avoid degeneracy and to ensure the exploration of the search space, as well. The pseudocode of the algorithm is introduced in Algorithm 1.

Algorithm 1 Pseudocode of the general PFO algorithm

```

1: Initialize the population by drawing  $N_s$  particles from the search space randomly and
   set all  $w_0^i$  weights to  $\frac{1}{N_s}$ . Set  $k = 1$ .
2: if  $k > 1$  then
3:     Move the particles according to the transition equation described by Equation 2.2
       or its modified form which will be defined later in Equation 2.9.
4: end if
5: if a decision variable is out of the boundaries then
6:     Continue with Row 2.
7: end if
8: for  $i \leftarrow 1$  to  $N_s$  do
9:     Compute the  $H(\mathbf{x}_k^i)$  objective value .
10: end for
11: Draw  $v_k$  observation noise from  $\varphi_k(\cdot)$ .
12: Calculate  $y_k$  observation according to Equation 2.3.
13: if  $y_k < y_{k-1}$  then
14:      $y_k = y_{k-1}$ 
15: end if
16: for  $i \leftarrow 1$  to  $N_s$  do
17:     Compute the  $w_k^i$  weight according to Equation 2.5.
18: end for
19: Execute resampling. (See the details in Section 2.2.2 and Appendix A.2.)
20: if stopping criterion is satisfied then
21:     Stop.
22: else
23:      $k = k + 1$ 
24:     Continue with Row 2.
25: end if

```

As in the case of PFO not a state estimation but an optimization problem is desired to solve, further explanation is required about the elements of the algorithm in this regard, which is provided with some additional considerations about the possible improvements of the general algorithm in Section 2.2.

2.2 Interpretation of the elements and data-driven improvements of the general particle filter optimization (PFO) algorithm

In this section, a more detailed description of the role of the algorithm elements is provided, and some novel ideas are proposed to improve the efficiency and investigate the objective function shape, as well.

2.2.1 Distribution of the observation noise

The distribution of the observation noise represented by $\varphi_k(\cdot)$ determines the way particles are differentiated during the re-weighting in the correction step of the PFO algorithm. This pdf firstly appears indirectly in Equation 2.3, and serves the definition of v_k observation noise that is drawn randomly at each iteration from $\varphi_k(\cdot)$. Thereby, the y_k observation can be generated according to Equation 2.3.

y_k is used to weight the particles in Equation 2.5 where $\varphi_k(\cdot)$ appears again. As the $\varphi_k(\cdot)$ function gives values higher than zero only on the domain of positive numbers by definition, it is clear that only those particles will have a positive weight whose performance is better than y_k ('good particles') and all the others zero ('bad particles') as illustrated in Figure 2.1. Thereby, only these particles with non-zero weights will propagate to the next iteration during the resampling step based on the $\pi_k(x)$ target pdf.

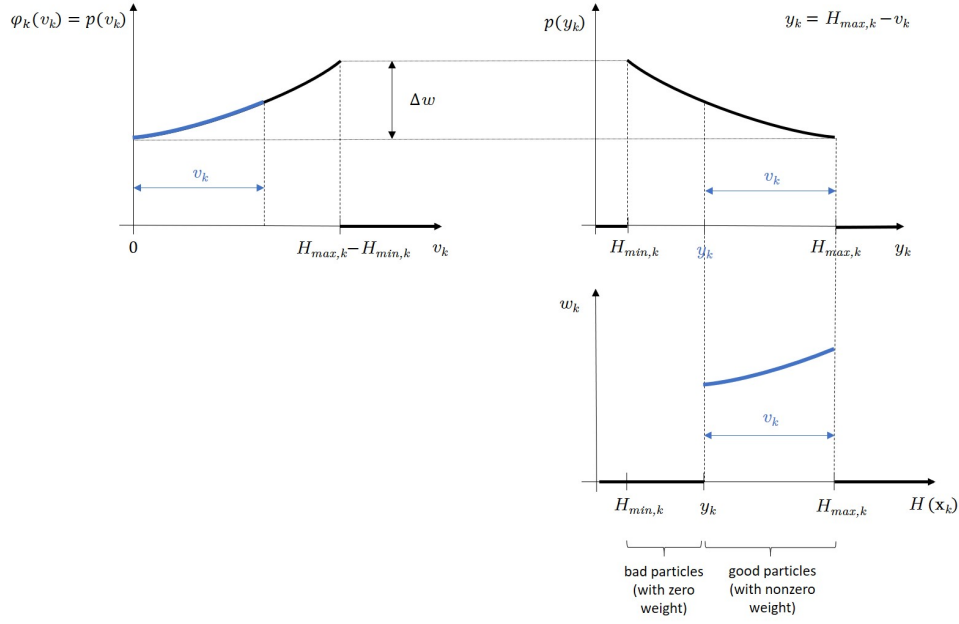


Figure 2.1: **Schematic figure of the connection between $\varphi_k(\cdot)$ density of v_k observation noise, y_k observation and the interval of the particle objective values in the k th iteration bounded by $H_{min,k}$ and $H_{max,k}$ and the particle weights (w_k) belong to them. $p(\cdot)$ refers to the probability density function of the variable in the argument. Δw is an adjustable parameter representing the steepness of $\varphi_k(\cdot)$.**

The question arises: how are the particles with non-zero weights wanted to be differentiated? The user has the chance to set it by the definition of $\varphi_k(\cdot)$. If the weights of these particles are desired to set equal, a uniform distribution is suggested to use, which is the most common case in the literature [13, 14]:

$$\varphi_k(v_k) = \begin{cases} \frac{1}{H_{max,k} - H_{min,k}}, & \text{if } 0 \leq v_k \leq H_{max,k} - H_{min,k} \\ 0, & \text{otherwise} \end{cases} \quad (2.6)$$

where $H_{max,k}$ and $H_{min,k}$ are the highest and lowest objective function values in the current particle set of the k th iteration.

However, to the very good particles, higher weights may be wanted to give than the less but still good ones to represent the different performances of them. In this case, the exponential distribution can be a possibly satisfying choice such as the one in Figure 2.1.

$$\varphi_k(v_k) = \begin{cases} \frac{\lambda_k e^{\lambda_k v_k}}{\int_0^{H_{max,k} - H_{min,k}} \lambda_k e^{\lambda_k v_k} dv_k}, & \text{if } 0 \leq v_k \leq H_{max,k} - H_{min,k} \\ 0, & \text{otherwise} \end{cases} \quad (2.7)$$

where

$$\lambda_k = \frac{\log \Delta w}{H_{max,k} - H_{min,k}} \quad (2.8)$$

where Δw is adjustable representing the weight ratio of the best and worst particle in the current particle set if all the particles would be kept ($y_k = H_{min,k}$), i.e., it refers to the steepness of the function.

In Section 3, these two cases will be compared to see which is the most suitable choice based on the search performance.

2.2.2 Proposals to ensure the exploration of the search space

Similarly to PF, the so-called sample impoverishment problem is also an issue that has to be handled in the case of PFO. As the new particles are chosen from the old ones during the resampling step, after a few iterations, the particles would concentrate on a few points of the search space (which would even be points from the initial population) [12]. Thereby, the exploration of the search space would be very poor [13].

There are two main directions in handling this issue.

- (a) Application of a more *complex resampling strategy* [13].
- (b) Introduction of a *transition kernel* [14].

The first one usually means that during the resampling, not the old particles are used to draw the new ones from, but a continuous probability density fitted to them. This can be, e.g., a Gaussian mixture distribution such as in the PE-SMC method [7]. The drawback of this strategy is that the fitting can be difficult in multidimensional spaces. On the other hand, the shape of the pdf should be approximated that can be a complex task in every iteration such as in the aforementioned PE-SMC, that is very time consuming.

A much simpler solution for the problem is the usage of a transition kernel besides resampling with replacement (where every particle is chosen with a probability proportional to its normalized weight.). It can be considered as a noise on the particles after the resampling which serves to move them away in their vicinity. The modified version of Equation 2.2 is defined as:

$$\mathbf{x}_k = \tilde{\mathbf{x}}_{k-1} + \mathbf{u}_k \quad (2.9)$$

where $\mathbf{u}_k$ refers to the random variables introduced to diversify the particles, drawn from a transition kernel that is symmetric to zero. (It has to be noted that the modified state transition described by Equation 2.9 is analogous with the one of the PF where $\mathbf{u}_k$ represents the process noise. In this case, the sample impoverishment problem occurs when the variance of this noise is too small [12], and can be solved, e.g., by implying a genetic-algorithm-based improvement of the PF [15].)

At the beginning of the optimization, higher $\mathbf{u}_k$ noise can ensure the exploration of the whole search space. At the same time, a lower one is more suitable as the iterations progresses, in order to not move the particles out from the areas with high objective value. It was declared in [14], that a gradually decreasing $\mathbf{u}_k$ trajectory is preferable. However, it was not fixed, which kind of trajectory is the most beneficial.

For the above-mentioned reasons, a comparative convergence analysis besides different $\mathbf{u}_k$ trajectories is carried out in this study. Three trajectories are examined. The elements of $\mathbf{u}_k$ are drawn from a zero-mean multivariate normal distribution in every iteration with variances

- (a) $\frac{x_{max}-x_{min}}{\alpha k}$, linear trajectory
- (b) $\frac{x_{max}-x_{min}}{\alpha k^\beta}$, exponential I.
- (c) $\frac{x_{max}-x_{min}}{\alpha \beta^k}$, exponential II.

where x_{max} and x_{min} refer to the upper and lower limits of the decision variables, respectively, α and β are parameters that has to be tuned, and k denotes the iteration number.

In terms of the extended algorithm, it is crucial to plan the shape of the $\mathbf{u}_k$ trajectory. The above-mentioned three strategies to generate the $\mathbf{u}_k$ trajectory will be compared and analysed according to their performance in finding the global optimum in Section 3. As for the resampling strategy, the one called systematic resampling is used in this work that is introduced in detail in Appendix A.2.

2.2.3 Stopping criterion issues

As for the end of the algorithm, the question arises how to define the stopping criterion. Liu used the one that in the last ten iterations no better solution is found [7]. It only involves the information about the best particle of the sample, however, there may be a demand to contain information about the whole particle set.

From this viewpoint, it seems a more suitable idea to define the termination condition according to the expected value of the particles considering the objective value of them. If the change of this value decreases under a tolerance value (δ), this can indicate that the distribution of the particles scarcely changes, thus the optimization can terminate.

$$|E(H(\mathbf{x}_k)) - E(H(\mathbf{x}_{k-1}))| \leq \delta \quad (2.10)$$

As in this work the distribution of the particle is also wanted to analysed, I suggest this condition which seems more suitable to achieve the goals were set. Its performance will be compared to that of the stopping criterion suggested by Liu in Section 3 [7].

2.3 A clustering-based method for exploring the shape of the objective function

After getting the final set of particles, the question arises how to evaluate the results. The global optimum can be defined as the particle with the best performance in the last iteration, or along the iterations [7]. The mean of the particles according to their position can be another satisfactory choice, however, only in those cases when it is sure that all the particles are gathered around that one point of global optimum. On the other hand, the structure of the PFO algorithm implies that the particles are concentrated around not one but a few points of the search space: the global and some local optimums. If these local optimums wanted to be found, a method is needed to sort the particles according to their position, and then the best values of the groups can be chosen.

A promising solution for the task is using a clustering algorithm. The most commonly used one is the K-means algorithm which divides the space of the data into K non-overlapping clusters and thus belongs to the group of partitional clustering methods. K-means can also be categorized as a prototype-based or center-based method because of the cluster definition of it: the points forming a cluster are closer to the center of their own cluster than any other [16].

During clustering, the centroids of the groups ($\{\mathbf{c}_i\}_{i=1}^K$) are moved and the data points are assigned to the currently closest centroid. In fact, the aim is to solve the following optimization problem:

$$\min_{\{\mathbf{c}_j\}_{j=1}^K} \sum_{j=1}^K \sum_{\mathbf{x}_k^i \in C_j} d(\mathbf{c}_j, \mathbf{x}_k^i) \quad (2.11)$$

where $\mathbf{c}_j$ represents the centroid of the j th cluster which elements are gathered in the C_j set ($j = 1, \dots, K$), and $d(\cdot, \cdot)$ refers to the distance measure.

The pseudocode of the algorithm can be seen in Algorithm 2.

Algorithm 2 Pseudocode of the K-means method [17].

- 1: Choose the number of clusters (K).
 - 2: Define the initial centroids.
 - 3: Assign the data points to the closest centroid considering squared Euclidean distance metric.
 - 4: Recompute the centroids as the mean of the data points in the cluster.
 - 5: **if** the centroids are the same as the previous iteration **then**
 - 6: Stop.
 - 7: **else**
 - 8: Continue with Step 3.
 - 9: **end if**
-

As seen in Algorithm 2, the number of the clusters is a user-defined property. However, if the method is wanted to be applied for the evaluation of a particle set where the number of clusters is not known, the optimal value of it can be determined by the so-called *gap statistic* which was shown to outperform other methods (e.g. Calinski-Harabasz, Hartigan, Silhouette) in case of high dimensions or elongated clusters [18].

It is worth using the method described above to evaluate the final and the intermediate particle sets and their propagation, as well. By clustering the particle sets, information can be gained about the relation of the decision variables and the objective value: along which variables can be recognised more groups, and which ones concentrate around one point. It can also be explored how the variables converge one by one, and the PFO algorithm may be improved further according to this information. The complexity of the objective function can also be characterized by the gained information.

As it can be presumed, the number of the clusters in the particle set, i.e., the found local optimums can be influenced by some factors of the algorithm, e.g., the $\varphi_k(\cdot)$ density or the sample size (N_s). It was also the subject of my investigations to discover which factors have a significant effect on it, and how they can be defined according to the goals of the user.

The above-mentioned issues are investigated on the chosen benchmark function, and the results are introduced in Section 3.

2.4 The extended form of the PFO algorithm

The issues mentioned in Section 2.1-2.3, thus the methodological background of the work is summarized in Figure 2.2.

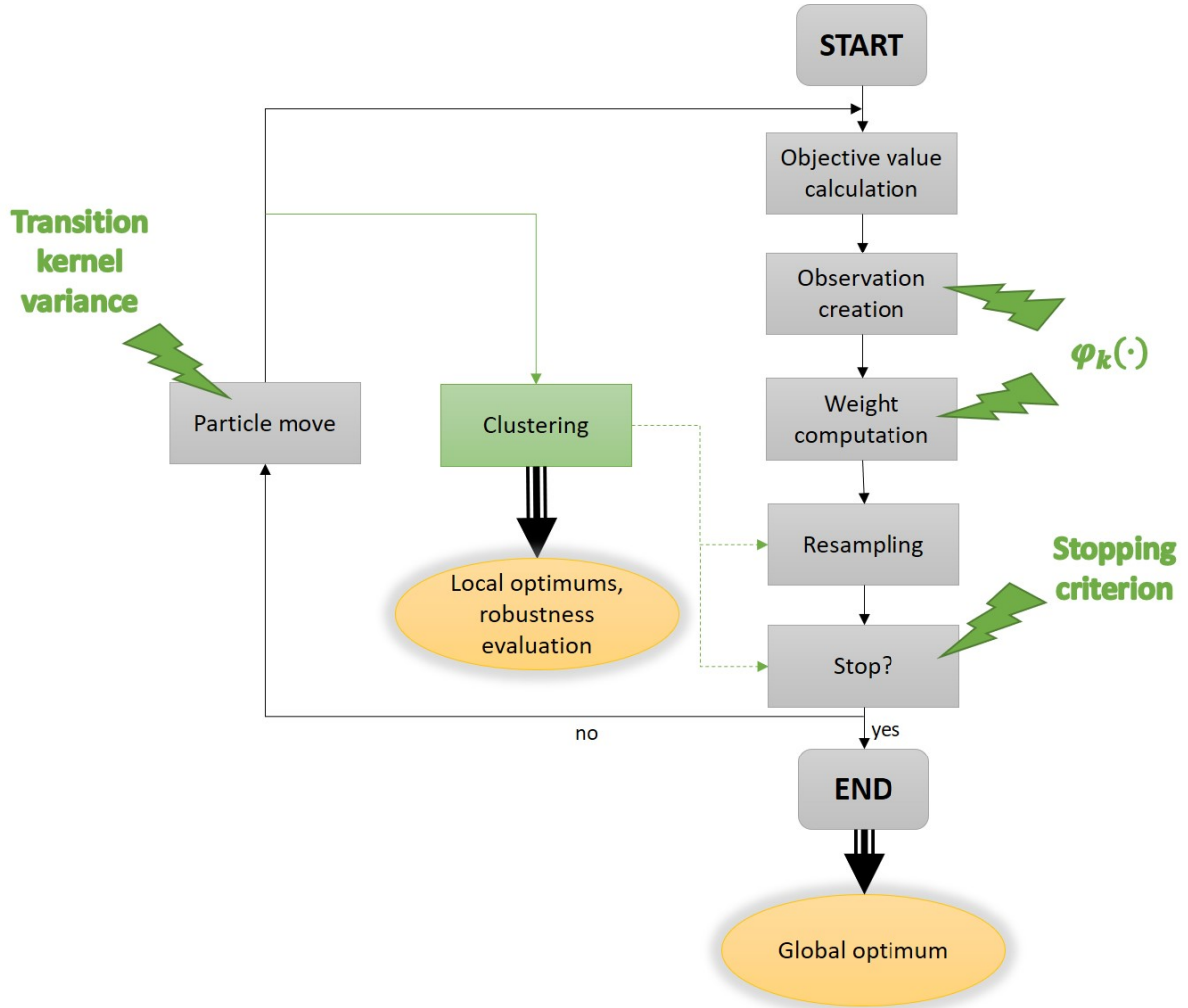


Figure 2.2: **Summary of the extended algorithm.** The elements where improvements are proposed are marked by green. The outputs of the general and extended algorithms are presented by yellow. The possible future research directions to develop the algorithm according to the clustering results appear as dashed lines.

The factors where modifications are introduced to the general PFO algorithm are seen by green such as the clustering-based extension in Figure 2.2. During the experiments, the effect of the type of the transition kernel variance trajectory, $\varphi_k(\cdot)$ and the stopping criterion to the global optimum finding performance is analysed.

The clustering step functions as a post-evaluation part here that investigates the gained particle set during the iterations, thereby, exploring the shape of the objective function. Evaluation of the robustness of the found local optimums is also possible with this technique. Although this work does not deal with this issue, the resampling strategy or the stopping criterion could be even developed according to the results as will be discussed later. These potential research directions are marked by dashed lines in Figure 2.2.

Chapter 3

Optimisation experiments on the Michalewicz-function

The experiments aimed to discover the potentials of PFO in the exploration of the search space and the shape of the objective function were executed on a benchmark function that was also used to evaluate the performance of PFO in [7], thus ensuring comparability. My choice was the Michalewicz function which can be flexibly tuned regarding its dimension and the difficulty of the search determined by the steepness parameter, as well.

The mathematical expression of the function can be written as:

$$f(\mathbf{x}) = \sum_{i=1}^{N_d} \sin(x_i) \sin^{2m} \left(\frac{ix_i^2}{\pi} \right) \quad (3.1)$$

where N_d denotes the dimension of the function and m is the steepness parameter.

The function is originally in a negative form, however, the positive one is used here as a maximisation problem is wanted to be defined. The function has $N_d!$ local optimums and the $m = 10$ value is used similarly to [7]. The domain is $x_i \in [0, \pi]$.

In this section, the tuning and monitoring of the PFO algorithm and its results are introduced through experiments on the aforementioned Michalewicz function defining a difficult search with many local optimums by conducting investigations on the following research questions as guidelines:

Research Question I. Is it worth differentiating the weights of the 'good particles', or taking equal weights is the best strategy to propagate the sample elements?

Research Question II. Which trajectory of the transition kernel is the most suitable choice to ensure the satisfying performance of the algorithm?

Research Question III. Which kind of stopping criterion is better to define the termination of the algorithm?

Research Question IV. How efficiently can the distribution of the particles be anal-

used by the clustering technique?

Research Question V. How do the clusters form and propagate during the iterations? What information can be gained from it regarding the complexity of the objective function?

Research Question VI. Which factors affect the number of clusters? How can they be influenced according to the aims of the user?

The chosen benchmark objective function is suitable for the determined research aims as it has $N_d!$ local optimums besides the global one, and it can be parameterized flexibly.

3.1 Analysis and improvements of the efficiency of the PFO algorithm for global search

At first, I aimed to investigate the possible trajectories of the transition kernel variance, $\varphi_k(\cdot)$ probability densities and stopping criteria of the algorithm and define which variants can be the most suitable for global search aims.

The experiments on the transition kernel were executed under the conditions defined in [11] for comparability. Namely, $\varphi_k(\cdot)$ was taken uniform, the sample size (N_s) is 2000 in the 5D and 5000 in the 10D cases, the algorithm stops if in the last 10 iterations no better solution was found, and 100 independent runs are executed. In Table 3.1, the means and standard deviations of the results of the 100 runs are presented while examining the trajectories of the transition kernel variance. The comparison of the different trajectories was made in the 5D case. The three cases (one linear and two exponential) defined in Section 2.2.2 and a constant one were examined besides tuned parameters in each case.

Table 3.1: Optimization results on the 5D Michalewicz function besides different variance trajectories of the transition kernel. The tuned parameters of the different types of trajectories (α, β) defined in Section 2.2.2, the means (μ) and standard deviations (σ) of the optimal objective function value ($H(\mathbf{x}^*)$) and the number of iterations was needed to the convergence (M) of 100 independent runs are introduced in the table. The objective function value of the real global optimum is 4.687658.

Variance trajectory			$H(\mathbf{x}^*)$ [-]		M [-]	
Type	α	β	μ	σ	μ	σ
constant	35.0	-	4.41	0.25	28.97	7.16
linear	3.0	-	4.59	0.10	44.91	12.98
exponential I.	1.5	1.65	4.61	0.09	49.75	15.39
exponential II.	4.5	1.10	4.57	0.10	102.52	71.53

The results confirm that besides a constant transition kernel, the performance of the PFO is worse than in the case of a decreasing trajectory. The linear and the two exponential trajectories bring similar results in the optimal objective function values. However,

the iteration number of exponential II. trajectory is markedly larger than the other two. Consequently, the linear and exponential I. trajectories are offered to choose in this case. In the further experiments, the exponential I. will be used.

Secondly, the issue of the $\varphi_k(\cdot)$ probability density was examined. On the one hand, it affects the sampling of the v_k observation noise thus the position of y_k between $H_{min,k}$ and $H_{max,k}$. Consequently, the ratio of the 'bad' and 'good particles' is also affected. On the other hand, it also determines how the weights of the 'good particles' are distinguished: whether equal weights are assigned to them or they are differentiated, and what the degree of this differentiation is. The illustration of these effects can be seen in Figure 2.1.

Table 3.2 contains the results besides uniform and exponential distribution of $\varphi_k(\cdot)$ as defined in Section 2.2.1, with different steepness of the slope in case of the latter which can be controlled by the Δw value. The higher the Δw is, the steeper the exponential function is, and the more the weights of the 'good particles' are differentiated.

Table 3.2: **Experiments on the Michalewicz-function besides different types of $\varphi_k(\cdot)$.** The steepness of the exponential distribution can be governed by Δw , as defined in Section 2.2.1. The means (μ) and standard deviations (σ) of the optimal objective function value ($H(\mathbf{x}^*)$) and the number of iterations was needed for the convergence (M) of 100 independent runs are introduced in the table. The objective function values of the real global optimums are 4.687658 and 9.660150 in the 5D and 10D cases, respectively.

		5D ($N_s = 2000$)				10D ($N_s = 5000$)			
		$H(\mathbf{x}^*)$ [-]		M [-]		$H(\mathbf{x}^*)$ [-]		M [-]	
Type of $\varphi_k(\cdot)$	Δw	μ	σ	μ	σ	μ	σ	μ	σ
uniform	-	4.6076	0.0915	49.75	15.39	8.8928	0.4981	70.97	24.57
exponential	3	4.6322	0.0690	53.20	19.27	8.8808	0.3932	71.58	34.40
exponential	10	4.6583	0.0554	55.97	18.47	9.0632	0.2662	71.02	21.38
exponential	30	4.6629	0.0436	57.44	19.85	9.1004	0.2797	80.04	24.11
exponential	100	4.6682	0.0666	59.73	17.30	9.146	0.2259	80.76	23.08
exponential	300	4.6725	0.0299	60.95	19.74	9.1904	0.2116	83.20	27.05
exponential	1000	4.6714	0.0238	58.17	18.48	9.1971	0.1861	87.66	28.98

According to the results in Table 3.2, it is clearly visible that the exponential distribution gets closer to the global optimum and thus performs better besides almost all parameter values than the uniform one at the expense of a not-so-significant growth in the required iteration number. A trend caused by the increase of Δw has to be also considered: the mean of the found optimal solutions ($H(\mathbf{x}^*)$) in the 100 runs increases, while the standard deviation of it decreases, and the mean of the required number of iterations (M) increases besides similar standard deviations.

The tendency of M is caused by the fact that as the steepness of the slope grows, larger v_k -s (thus smaller y_k -s) are more often sampled. Thereby, a higher proportion of the particles get into the 'good particle' groups, and even though the worse ones

get smaller weights, more iteration is needed. However, it can be concluded from the tendency of $H(\mathbf{x}^*)$ that the weights of the 'good' particles are worth differentiating as better optimization performance can be reached.

It has to be noted that even if in the 5D case the real optimal solution was approached well enough, in the 10D problem no. The reasons of this phenomenon are, on the one hand, that not only the dimension of the search space grow significantly, but also the number of local optimums that is $10!$ in this case. Although the number of particles was raised, it seems an even larger one would be more preferable.

The stopping criteria were also examined, and the results are collected in Table 3.3.

Table 3.3: Comparison of the performance of different stopping criteria on the 5D Michalewicz function. δ refers to the tolerance value defined in 2.2.3. The means (μ) and standard deviations (σ) of the optimal objective function value ($H(\mathbf{x}^*)$) and the number of iterations was needed for the convergence (M) of 100 independent runs are introduced in the table. The objective function value of the real global optimum is 4.687658.

Stopping criterion according to the...	$H(\mathbf{x}^*)$		M	
	μ	σ	μ	σ
best particle of the last 10 iteration [7]	4.6725	0.0299	60.95	19.74
expected value of $H(\mathbf{x})$ in the sample ($\delta=0.01$)	4.6590	0.0365	25.62	3.73
expected value of $H(\mathbf{x})$ in the sample ($\delta=0.001$)	4.6687	0.0328	39.60	6.04
expected value of $H(\mathbf{x})$ in the sample ($\delta=0.0001$)	4.6707	0.0273	54.18	11.77

As seen in Table 3.3, the stopping criterion defined by the expected value brings a similarly satisfying performance as the other defined by the best particle of the last ten iterations, besides a lower iteration number. It can be concluded that in large-scale simulations or cases when the speed of the optimization is a crucial issue, the expected value-based stopping criterion can be used more efficiently without losses on the performance.

Comparing the convergence results with those in [11] shown in Table 3.4, it can be concluded, that my proposed improvements bring significantly better performance than the general PFO algorithm by managing and developing the existing elements of the algorithm without implementing a computationally expensive extension e.g. in PE-SMC.

Table 3.4: Comparison of the performance of the PFO containing my improvements and the results found in the literature about that of the general PFO and other algorithms (PSO, PE-SMC) [11]. The table contains the mean of the optimal objective function values of 100 runs in all four cases.

N_d	$H(\mathbf{x}^*)$	PFO with my improvements	General PFO	PSO	PE-SMC
5	4.687658	4.6725	4.2103	4.6624	4.6875
10	9.660150	9.1971	5.9357	9.4562	9.6596

This section aimed to investigate the performance of the proposed variations of the general PFO algorithm in finding the global optimum of a complex benchmark function

with many local optimums defining a difficult search. As was seen, the performance can be improved by the use of a decreasing trajectory of the transition kernel variance and by differentiating the particle weights by a suitable choice of $\varphi_k(\cdot)$. On the other hand, the change of the stopping criterion brought some advantages in the computational time by reducing the number of required iterations. Section 3.2 will explore how further information about the shape of a complex objective function can be gained from the propagated particle set.

3.2 Clustering-based technique for finding local optimums

The PFO gives not only the estimation of the global optimum as a result but also a particle set, i.e., a discrete distribution representing the objective function values. This additional information makes it possible to investigate the local optimums of the function.

The structure of the PFO algorithm results that the particles propagate in time and concentrate around the more promising areas, i.e., the local optimums of the function. An example of the distribution of the particles during the different phases of the search represented by the k iteration number can be seen in Figure 3.1.

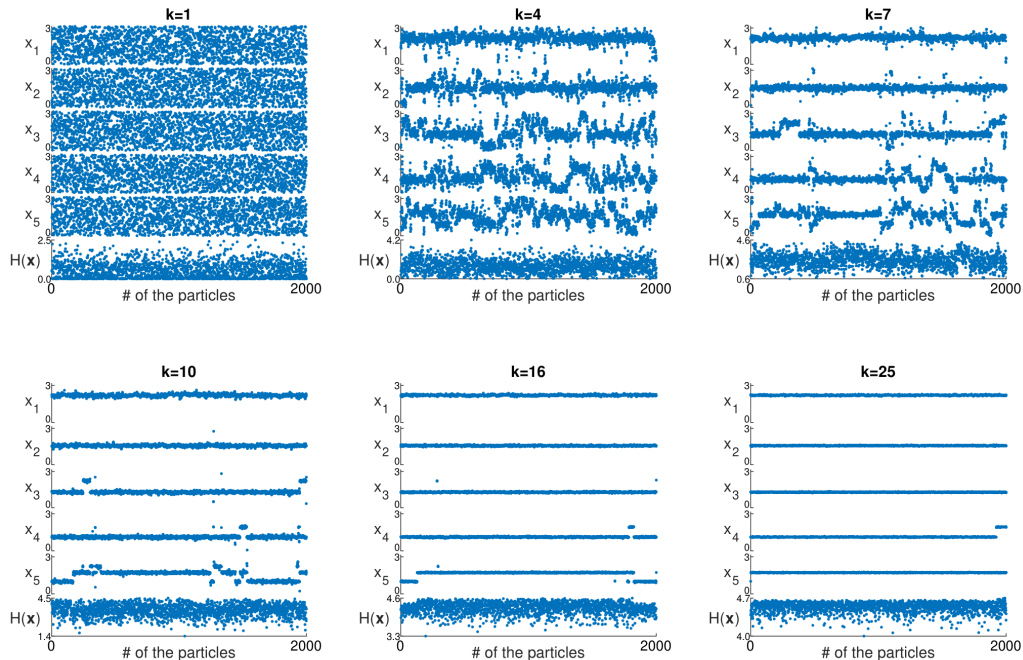


Figure 3.1: **Propagation of the particles during the first 25 iterations.** The experiment was executed on the 5D Michalewicz function with exponential $\varphi_k(\cdot)$ and $N_s = 2000$.

It is seen in Figure 3.1 that the particles cover the whole search space at the beginning ($k = 1$), then they start to concentrate in some areas of it and finally converge around one point. It is also seen that the perturbation of the particles gets smaller during the iterations as the variance of $\mathbf{u}_k$ is decreased gradually.

The sensitivity of the global optimum to the decision variables can also be investigated according to Figure 3.1. Considering the coordinates of the particles in the current sample, x_1 and x_2 decision variables converge, i.e., concentrate in the neighborhood only one point faster than the other three which is also seen in Figure 3.2 illustrating the particle with the best performance. Thereby, it can also be concluded that the global optimum is more sensitive to these two variables than the others.

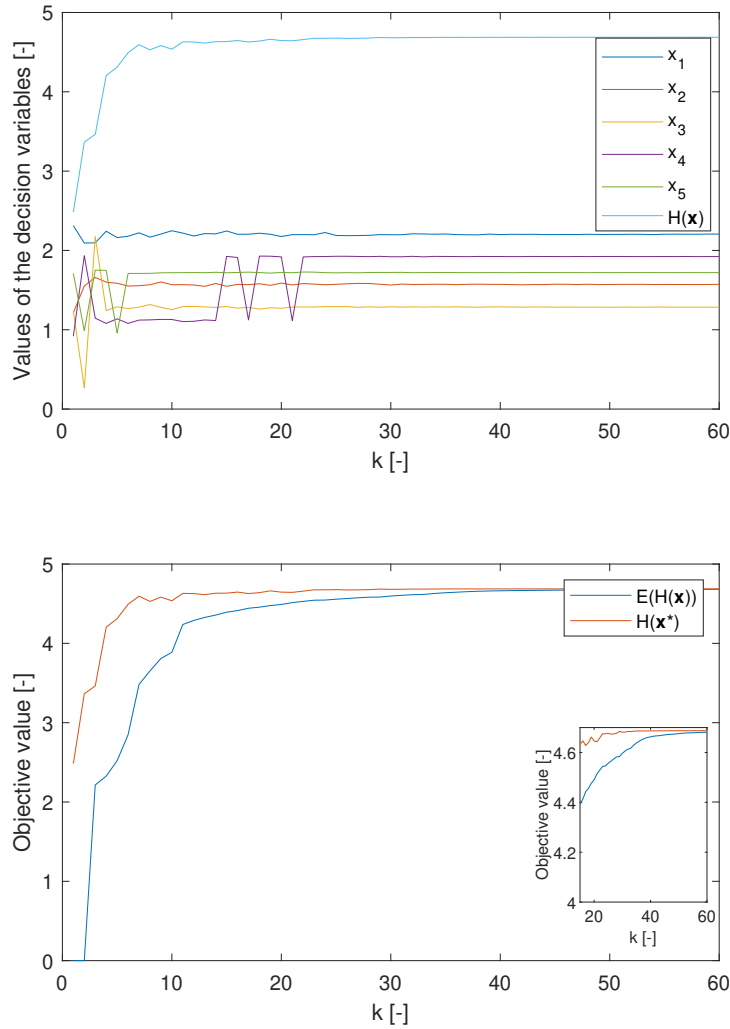


Figure 3.2: **Convergence results belonging to the experiment represented in Figure 3.1.** The coordinates of the best particle in the current population, the objective value belonging to the best particle ($H(\mathbf{x}^*)$) and the mean objective value of the sample set ($E(H(\mathbf{x}))$) along the iterations are illustrated.

The convergence of the objective function value of the best particle ($H(\mathbf{x}^*)$) and that of the mean of the particle set can also be seen in Figure 3.2. The convergence of the mean is slower than that of the best particle as even the particles with lower performance are considered, and on the other hand, the decreasing perturbation on the variables also affects to the convergence in that case. Consequently, the idea of using the change of the best particle as a stopping criterion instead of the mean of the particle set may arise which is suitable if the aim is only to find the global optimum and there is only one global optimum. However, if the objective function has more global optimums with equal performance, or the local optimums are investigated, the mean value can be a more suitable choice to conduct the experiments.

It is seen in Figure 3.1 that the particle set tends to split into well-distinct groups after a few iterations. As the particles concentrate on the places with high objective values, the local optimums can be identified by the analysis of these groups. The particles can be sorted by K-means clustering, thereby, the centers of the clusters can be identified, as well. For example, the clustering results of the particle set in the 16th, 25th, 30th and 35th iteration are introduced in Figure 3.3.

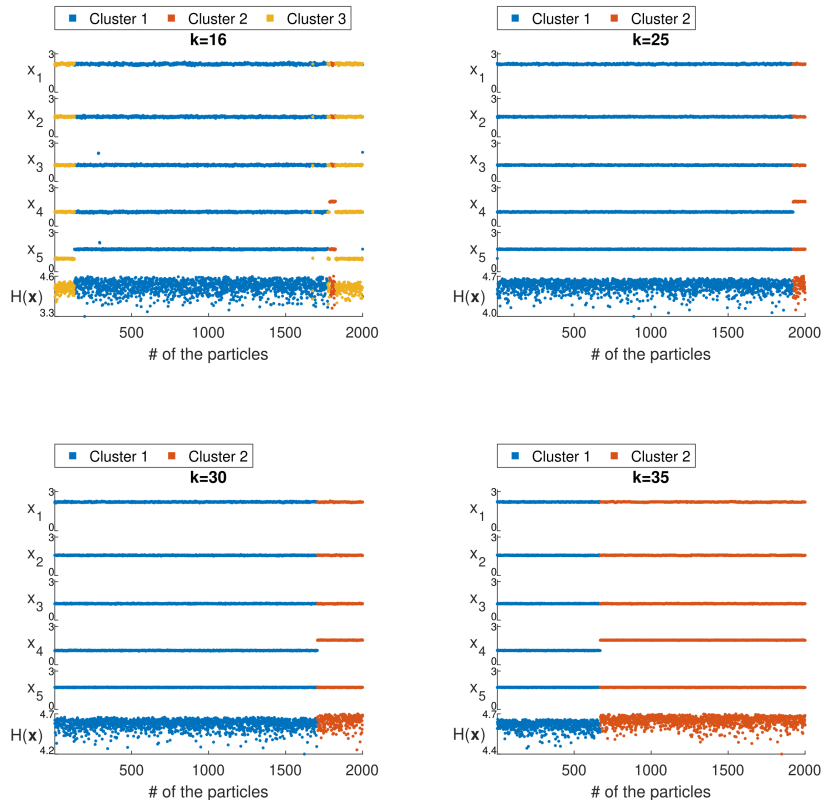
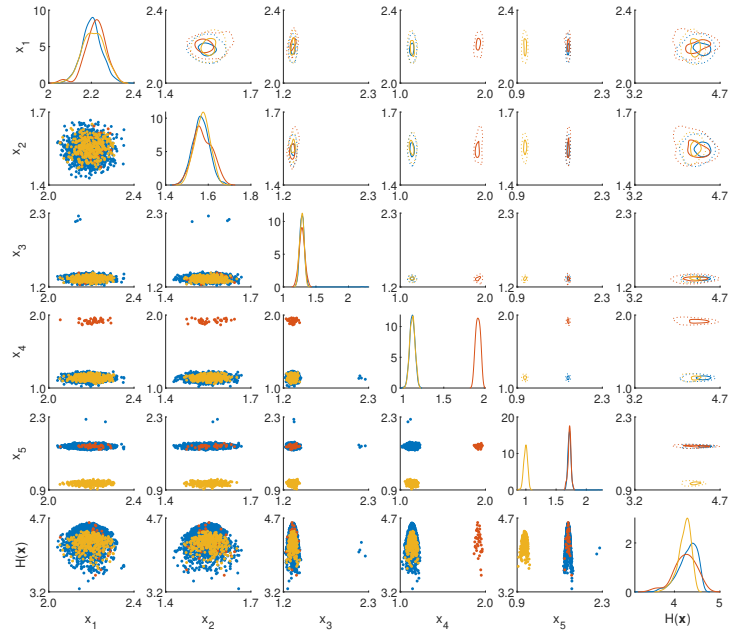


Figure 3.3: **Evolution of the clusters along the iterations.** Decision variable and objective function values belonging to the 2000 particles are illustrated by different colors according to the cluster they belong.

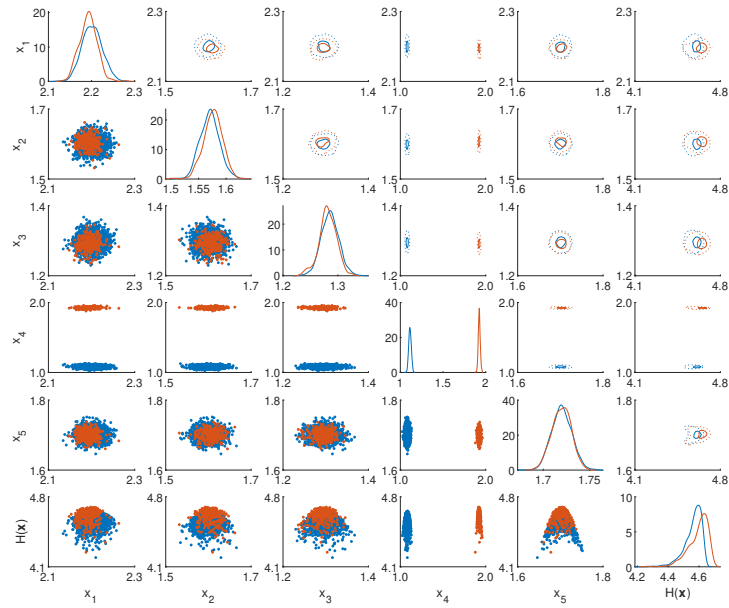
As seen in Figure 3.3, the particle set of the 16th iteration can be divided into three clusters, which means that the particles concentrate around three points of the search space, i.e., three local optimums of the objective function are found. It can also be noticed that the elements of the same cluster do not appear inordinately, but form groups according to the identifier number (#). The reason for this phenomenon is the chosen resampling strategy which is the so-called systematic resampling in my experiments presented in detail in Appendix A.2.

The particles of the cluster with the worst performance (Cluster 3) move to the other two groups, thereby, this cluster illustrated by yellow in Figure 3.3 disappears for the 25th iteration. After the 25th iteration, the particles move forward from Cluster 1 (blue) to Cluster 2 (red) tendentially, which suggests that these particles concentrate around an optimum with better performance than in the case of Cluster 1. It is also seen in Figure 3.3, that the upper limit of the objective values in Cluster 2 is higher than that of Cluster 1.

The pairplot of the decision variables and the objective function values belonging to the particles (Figure 3.4) gives more information about the sample set: in this case, the distribution of the particles is examined.



(a)



(b)

Figure 3.4: **Pairplot of the decision variables and objective function values belonging to the particles in the (a) 16th and (b) 30th iteration.** Probability densities are presented in the main diagonal, scatter plots below and contour plots above it. In the contour plots, the upper quartile is represented by continuous lines and the lower quartile by dotted lines. The axis limits are adjusted according to the distribution of the points for visibility, thereby, different in the (a) and (b) parts of the figure. The colors match with those of Figure 3.3: Cluster 1 is marked by blue, Cluster 2 by red, and Cluster 3 by yellow.

Similarly to the representation in Figure 3.3, it is seen that Cluster 3 separates from Cluster 1 by the x_5 variable, while Cluster 2 by the x_4 , and that the highest objective values belong to the elements of Cluster 2. On the other hand, it can also be considered that the points of Cluster 1 cover a larger area than those of Cluster 2, which is also seen in some of the probability density functions (e.g., in the case of x_1 and x_4 in Figure 3.4 (b)). These observations suggest that a less sharp thus more robust peak belongs to Cluster 1 than to Cluster 2 which is supposed to represent the global optimum. These suggestions are also confirmed by the information in Figure 3.5.

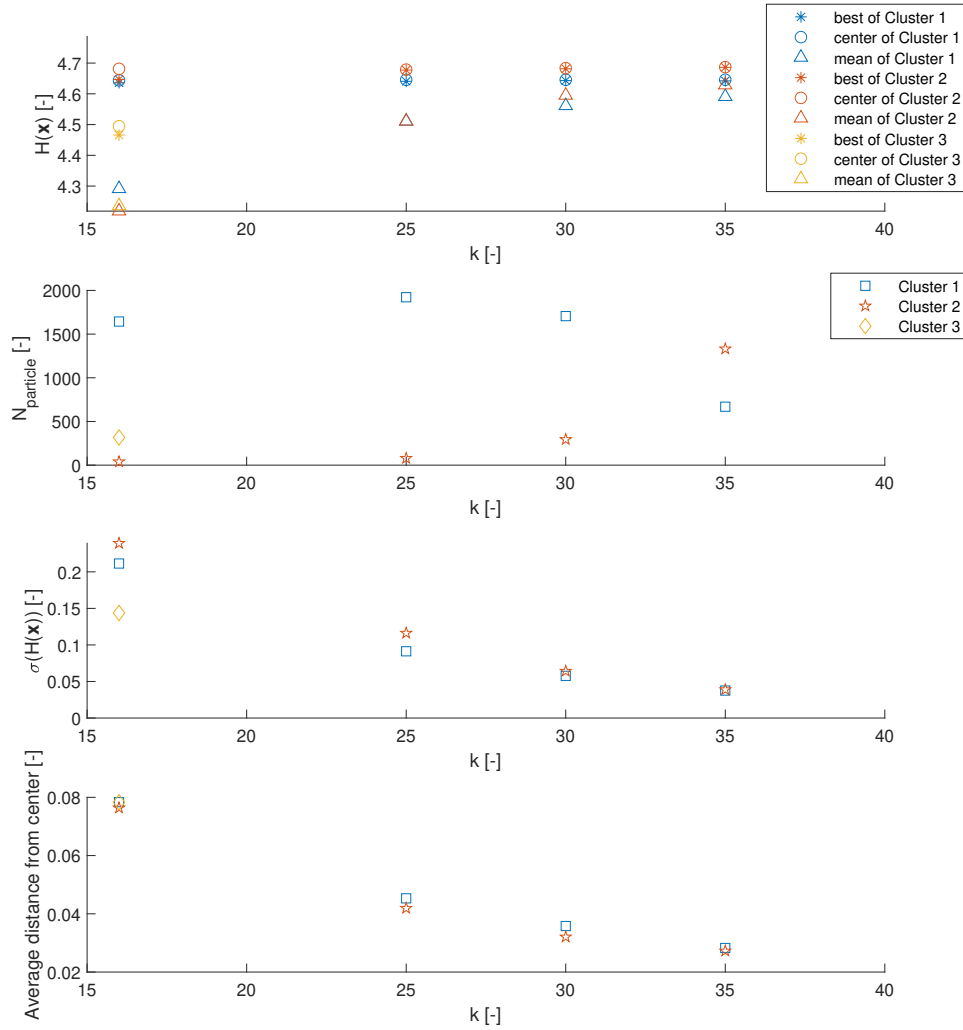


Figure 3.5: **Analysis of the properties of the clusters along the iterations.** The objective value ($H(\mathbf{x})$) of the best particle of the cluster (*), the center of the cluster (O), the mean (Δ) and the standard deviation ($\sigma(\cdot)$) of the particle objective values, the average distance of the particles from the center and the number of particles ($N_{particle}$) in each group are illustrated. The colors that represent the clusters are consistent with that of Figure 3.3 and Figure 3.4.

The first observation that can be made by Figure 3.5 is that the cluster centers are nearly constant along the iterations examined. It is also seen that the best elements of the cluster have a bit lower objective value than the centers at $k = 16$, however, then they converge towards and finally take the value belonging to the centers. This means that the top of the peaks are found by the clustering analysis already in the 16th iteration and before a particle gets to exactly that point. This anomaly can be explained by the fact that in the early iterations, the peaks are more rarely sampled, partly because a more extensive neighborhood of the optimums is covered in this phase, partly because the presence of the other clusters reduces the number of particles per cluster.

As for the robustness of the peaks, it can be seen in Figure 3.5 that the average distance from the cluster center is higher in the case of Cluster 1 than that of Cluster 2 which is consistent with the suggestion made by Figure 3.4 about the covered area of the particles and the pdf of the decision variables. The tendency of the mean objective function of the clusters in Figure 3.5 also confirms the presumption that the optimum represented by Cluster 1 is more robust than Cluster 2. It can be seen that at the beginning, the mean $H(\mathbf{x})$ of Cluster 1 is higher than that of Cluster 2, despite the fact that the top of that peak is lower which suggests that in this case, higher proportion of the particles takes on a high objective value around the top of the peak (as clearly seen in the pdf plot of $H(\mathbf{x})$ in Figure 3.4 (a)), i.e., this optimum is more robust. Along the iterations, the mean $H(\mathbf{x})$ grows more rapidly in the case of Cluster 2 and it even outperforms Cluster 1 as the interval of the possible objective values gets narrower (which is also seen on the decreasing tendency of $\sigma(H(\mathbf{x}))$) and the impact of the difference between the top values increases. The 1D schematic figure of this issue can be seen in Figure 3.6.

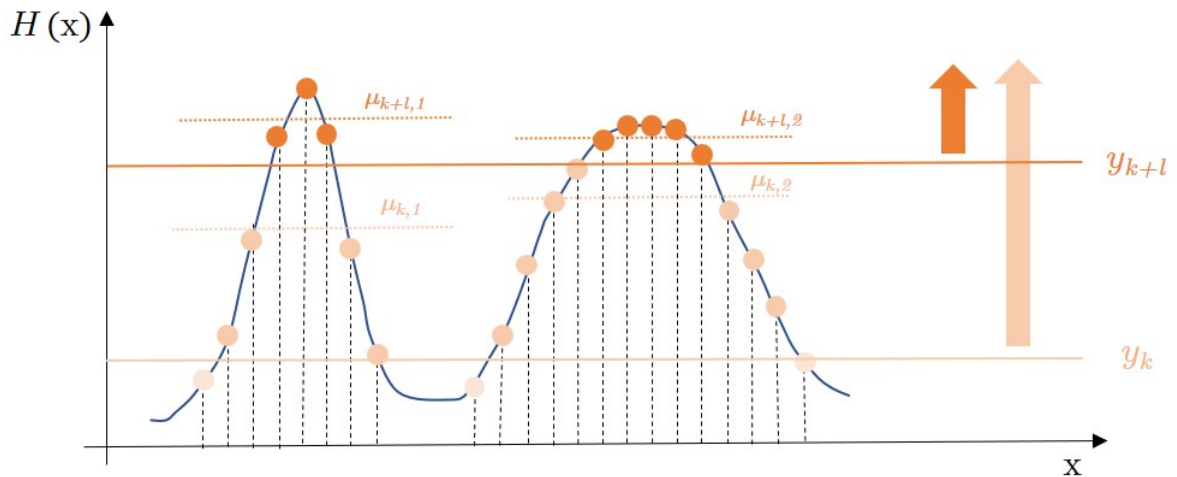


Figure 3.6: **Explanation of the change tendency of the mean objective values of the clusters in case of the global and a more robust local optimum shown in Figure 3.5 by a schematic representation in 1D.** $\mu_{k,j}$ denotes the mean objective value of the particles belonging to the j th peak, i.e., the j th cluster in the k th iteration.

The number of the particles in the clusters in Figure 3.5 also shows the tendency declared above, namely, Cluster 3 disappears and the particles propagate from Cluster 2 to Cluster 1. It can also be seen in Figure 3.5, that the particles from Cluster 3 propagate to Cluster 1 first, even though Cluster 2 has better performance. The reason for this phenomenon is, on the one hand, the large population size belongs to Cluster 1, and on the other hand, the higher robustness of this optimum.

In this section, it was introduced how PFO can be used to explore the complexity of the objective function. It was seen that the propagation of the particles carries information about the sensitivity of the objective function to the decision variables. It was also presented how the local optimums can be identified by the clustering of the particle set and how their robustness can be evaluated by the monitoring of the groups. In the next section, the factors that can affect the search and propagation of the particles are investigated.

3.3 Factors that affect the local optimum search performance

After discussing the information that can be gained by the analysis of the intermediate particle sets and the clusters formed from them, it is also investigated which factors of the algorithm have an effect on the search. In this section, two factors are examined: the sample size (N_s) and the distribution of $\varphi_k(\cdot)$.

The search was run besides four different N_s values, and the gained results can be seen in Figure 3.7. Similarly to the aforementioned experiments, the x_1 and x_2 variables were not defined clusters (concentrated around one point), thereby, only the x_3 , x_4 and x_5 decision variables are shown for visibility.

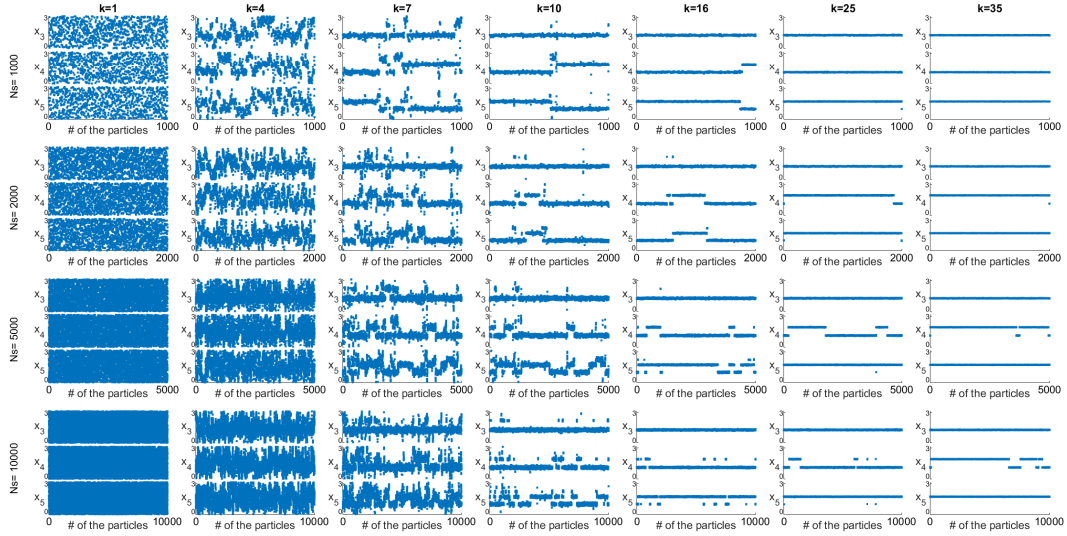


Figure 3.7: **Propagation of the x_3, x_4 and x_5 coordinates (decision variables) of the particles besides four different N_s values: 1000, 2000, 5000 and 10000.** The exponential version of $\varphi_k(\cdot)$ was used during the experiment.

It can be seen in Figure 3.7 that the number of the clusters does not depend on the sample size: the x_4 and x_5 variables form two-two groups in all the cases, and the x_3 only one. This means that it is not worth increasing the N_s if more local optimum is wanted to find to get more information about the complexity of the objective function. However, in the case of a too low sample size ($N_s = 1000$) it can be seen that the search can converge to a local optimum instead of the global one (the final value of x_4 is different than in the other three cases). For these reasons, the optimal N_s is the lowest value which is high enough to find the global optimum with high confidence.

It was also investigated if there is a difference between using exponential and uniform $\varphi_k(\cdot)$ in terms of finding local optimums. The results can be seen in Figure 3.8.

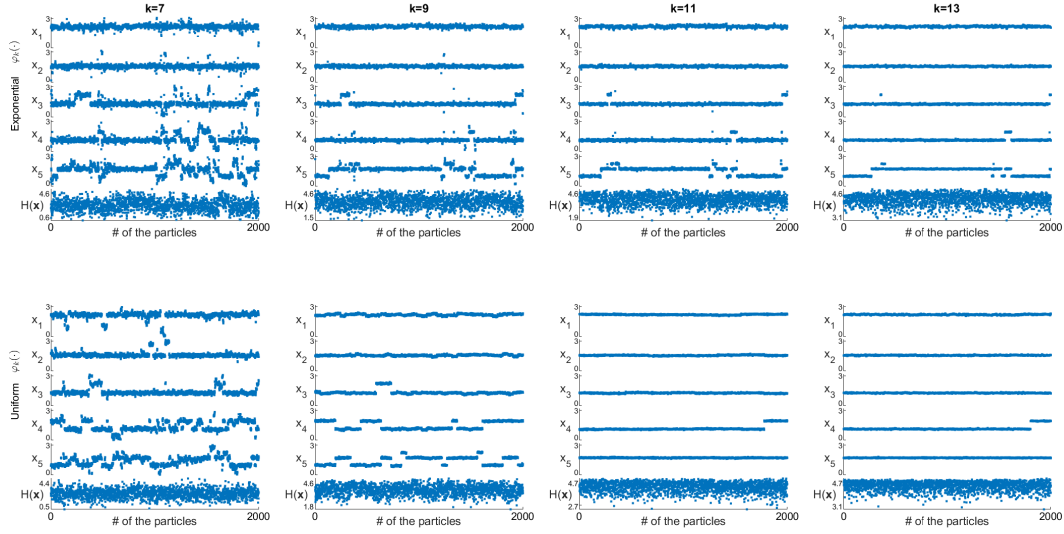
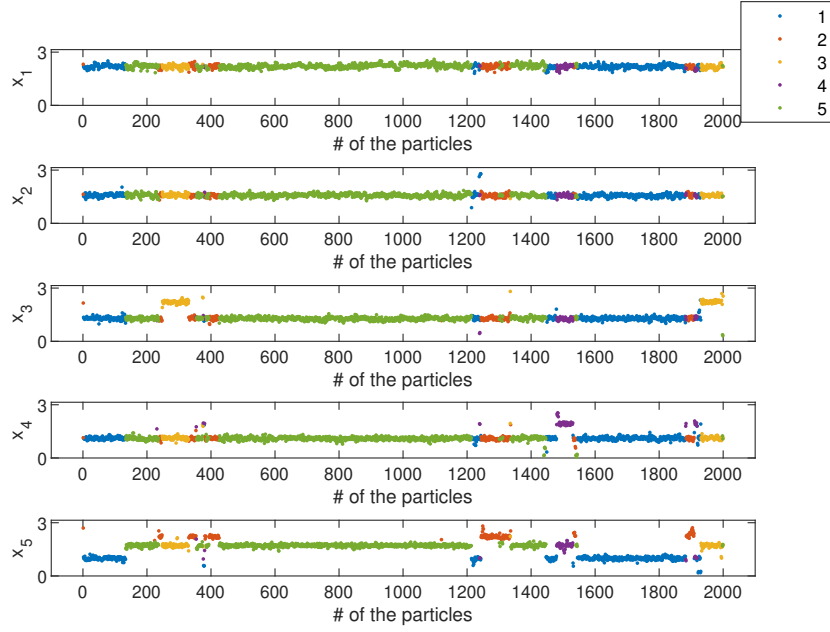
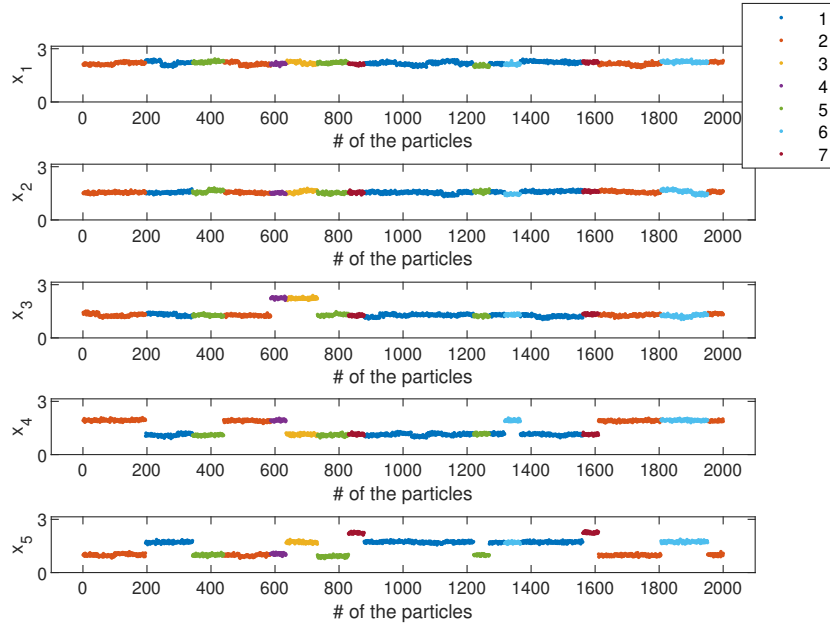


Figure 3.8: **Propagation of the coordinates (decision variables) of the particles besides exponential and uniform $\varphi_k(\cdot)$.** $N_s = 2000$ sample size was used during the experiment.

It is conspicuous in Figure 3.8, that the results besides uniform $\varphi_k(\cdot)$ give a clear view of the clusters more quickly along the iterations. On the other hand, the clusters are more well-separated than in the exponential case. Moreover, more cluster is found as also seen in Figure 3.9.



(a)



(b)

Figure 3.9: Clusters of the particle set of the 9th iteration can be seen in Figure 3.8 besides (a) exponential and (b) uniform $\varphi_k(\cdot)$.

The reasons for this phenomenon can be found in the different effects made by the $\varphi_k(\cdot)$ function on the search. As was seen in Figure 2.1, smaller y_k is more likely to be chosen in the exponential case, which implies that the level of y_k is elevating slower. This causes particles with low performance will be present for more time, thus the peaks can

be less distinguished as seen in Figure 3.10 marked by the red arrow.

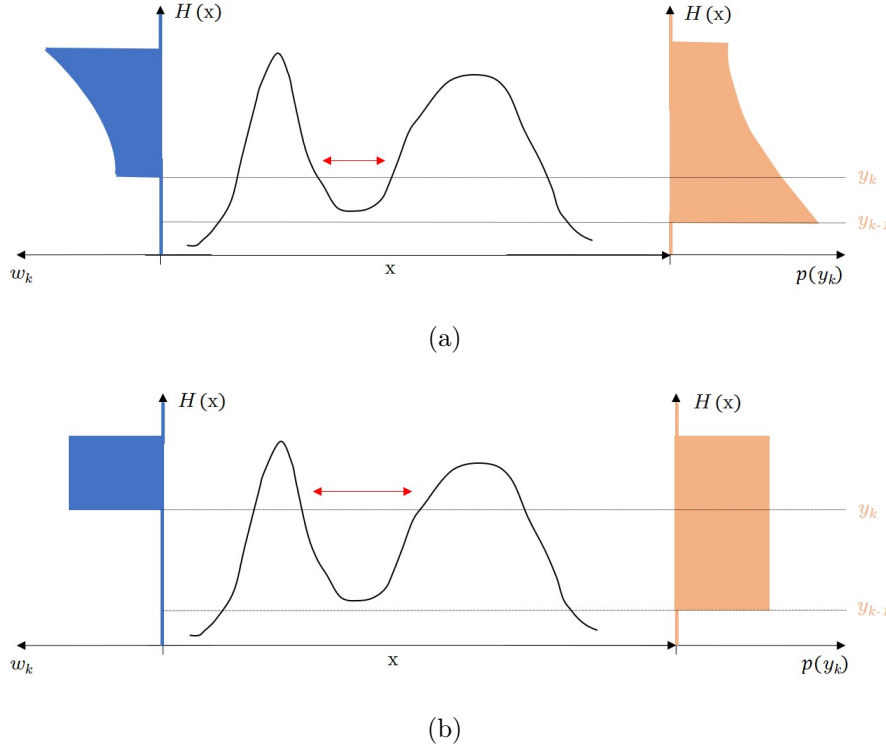


Figure 3.10: **Schematic representation in 1D for explaining the effects of $\varphi_k(\cdot)$ on the degree of separation of clusters in the (a) exponential and (b) uniform case.** On the left side of the figure, the distribution of the particle weights is presented schematically, and on the right side, the shape of the pdf of y_k is shown, representing the probability that the y_k level moves to a specific value.

On the other hand, these particles with poor performance get lower weights than the better ones as seen in Figure 3.10, thereby, despite the presence of more low-performance sample elements which causes the outliers, i.e., the less clear view of the clusters in Figure 3.9 (a), most of the particles will accumulate around the high-performance peaks. Thereby, when the particle set reaches a satisfying degree of separation to be able to form clusters, the particles are concentrated around the peaks with better performance, and the neighborhood of low-performance local optimums may be impoverished or evacuated even if they have higher performance on the top than the current y_k , which causes the less number of clusters, i.e., the less found local optimums. Meanwhile, in the uniform case, y_k is increased more rapidly which helps the separation of clusters, while all the 'good particles' are considered equal weights which gives the chance to find all the local optimums over the y_k level.

In this section, it was shown how the type of $\varphi_k(\cdot)$ and the population size (N_s) affect the exploration of a complex objective function. For the above-mentioned tendencies and phenomenons, if the aim of the user is to explore the local optimums and the shape of

the function, the application of the uniform $\varphi_k(\cdot)$ is suggested thus more optimums can be found, and the clusters representing them are more well-separated. Regarding N_s , the only requirement is to be large enough to explore the whole search space thus ensuring the finding of sharp peaks.

Chapter 4

Conclusions

This work investigated the efficiency of the PFO algorithm for global search and objective function shape exploration. Besides the detailed analysis and improvements of the general PFO algorithm, a clustering-based method was provided to discover the robustness of the peaks of the objective function by following the propagation of the particle set. This idea promotes to use the technique in problems where not the global but a more stable local optimum gives the best solutions for some practical reasons.

The elements of the general PFO algorithm were analysed and some recommendations were given to improve the global search performance by conducting a comparative study on the different variants. Firstly, it was shown that a decreasing transition kernel variance trajectory outperforms the constant one, and that a linear or the defined exponential I. trajectories are the most suitable choices to reach high performance besides low iteration number. Secondly, the advantage of the differentiation of the 'good particle' weights by an exponential $\varphi_k(\cdot)$ instead of a uniform one was shown. Finally, it was introduced how the stopping criterion defined by the expected value of the objective values belonging to the particles can give a similar performance to the one used before besides a lower iteration number. Some parameters (e.g., $\alpha, \beta, \Delta w$) of the PFO algorithm are problem specific, however, it some ideas were given how they can be tuned.

To investigate the complexity of the objective function, a clustering-based technique was proposed. By following the distribution of the particle set along the iterations, information was gained about the sensitivity of the optimums to the decision variables, the performance of the local optimums and their robustness which can be evaluated according to the top values, the average distance from the cluster center and the mean of the clusters. It was also declared that the N_s sample size does not have a significant effect on the performance of the proposed method, it only has to be large enough to converge to the global optimum. However, the choice of $\varphi_k(\cdot)$ influences the number of found optimums and the difficulty of the clustering, as well.

Considering the results it is clear that the favorable shape of $\varphi_k(\cdot)$ hardly depends on the aims of the user. If the goal is to find the global optimum with high confidence, the exponential one is suggested. If the purpose is to find local optimums thus exploring the shape of the objective function, the uniform variant is recommended.

The experiments were executed on the Michalewicz-function which defines a difficult and flexibly parameterized search with many local optimums. As most of the objective functions in the real world possess less complex shape, the results are promising in terms of moving the PFO from the theoretical to the practical level. Potential constraints often appear in real-world problems can also be easily implemented because of the structure of the algorithm.

One main drawback of the algorithm that appeared during the experiments is that its performance is somewhat weaker in high-dimensional cases. The problem can be remedied by larger population size or dimensionality reduction techniques, however, it must be kept in mind that the used benchmark function defines an extremely difficult search with sharp peaks and lots of local optimums (e.g. $10!$ in the 10D case) which is not typical in case of practical problems. Another limitation is, similarly to the population-based algorithms, the high computational demand. It was found, compared to the interior-point direct search algorithm that is commonly used in the engineering practise, that PFO does consume a bit more time, however, it is commensurable with that of the other besides the same performance. On the other hand, it was shown that the appropriate choice of the stopping criterion can further reduce it. Moreover, the individual convergence information of the decision variables gives the chance to fix the value of the converged ones thus excluding them from the further iterations and accelerating the algorithm.

In the future, a practical case study is planned to conduct where optimisation constraints also play a role. Experiments on the integration of the PFO optimisation and PF state estimation methods would be also interesting. Another possible research direction is the issue of how the results gained by the clustering-based method can be implemented into the search strategy. For example, as was seen, the cluster centers can define the optimums before the best particle converges there. This information can be used to reduce the iteration number, as well. Another possibility is to integrate the technique into the resampling strategy. These improvements can make the search and objective function exploration more efficient.

Appendix A

My Appendix

A.1 PF framework

The particle filter is originally a state estimation algorithm that is used to derive the unmeasured states of a dynamic system from the noisy measured ones even in nonlinear systems. It represents the states by a discrete probability density in the form of a particle set that is evolved. Every particle belongs to a point of the state space ($\mathbf{x}_k$) and owns a weight (w_k) which is updated in every time step. This set can be denoted in time k as:

$$p(\mathbf{x}_k | \mathbf{y}_k) \approx \sum_{i=1}^{N_s} w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \equiv \{\mathbf{x}_k^i, w_k^i\}_{i=1}^{N_s} \quad (\text{A.1})$$

where N_s marks the sample size and $\delta(\cdot)$ refers to the Dirac-delta function and $\mathbf{y}_k$ to the measurement data at time k .

The basic algorithm executes two steps iteratively. In the *prediction step*, all the particles are propagated according to the state transition equation of the model which uncertainty is also considered, thus $\mathbf{x}_{k+1}$ is calculated. In the *correction step*, the weights are updated according to the new measurement considering with what probability that exact measurement is gained if the system is in the $\mathbf{x}_k^i$ state, and also normalized which can be described in case of the i th particle as:

$$w_{k+1}^i = \frac{w_k^i \cdot p(\mathbf{y}_{k+1} | \mathbf{x}_{k+1}^i)}{\sum_{i=1}^{N_s} w_k^i \cdot p(\mathbf{y}_{k+1} | \mathbf{x}_{k+1}^i)} \quad (\text{A.2})$$

By these two steps, the distribution of the particles, i.e., the so-called posterior pdf at time $k + 1$ is gained:

$$p(\mathbf{x}_{k+1} | \mathbf{y}_{1:k+1}) = \{\mathbf{x}_{k+1}^i, w_{k+1}^i\}_{i=1}^{N_s} \quad (\text{A.3})$$

To avoid the degeneracy problem that means that a large proportion of the weights

concentrates on a few particles, in the general particle filter algorithm, a *resampling step* is also used: new, equal-weight particles are drawn from the discrete probability density defined by the old ones.

A.2 Systematic resampling

During the resampling step, every particle is chosen with a probability proportional to its weight.

Let us denote the cumulative sum of the weights of the first m particle as:

$$Q^{(m)} = \sum_{i=1}^m w_i \in [0, 1] \quad (\text{A.4})$$

where $1 \leq m \leq N_S$.

Systematic resampling is a type of resampling, where one number is chosen from each equidistant subinterval (which number is equal to N_s) that the $[0, 1]$ interval was divided into as:

$$z_j = z_1 + \frac{j-1}{N_S}, \quad j = 2, 3, \dots, N_S \quad (\text{A.5})$$

where z_1 is a random number from the uniform distribution on $(0, \frac{1}{N_s}]$.

After that step, the $Q^{(m)}$ ($m = 1, \dots, N_s$) numbers can also be marked on the $[0, 1]$ interval of the number line, and $\mathbf{x}_m$ is drawn for the new sample set as many times as many z_j satisfies the equation:

$$Q^{m-1} < z_j \leq Q^m \quad (\text{A.6})$$

For this reason, the new particles will be in the same order as the old ones, except that some particles will disappear, and some will multiply.

List of symbols

$\mathbf{x}$	vector of the decision variables $(x_1, x_2, \dots)$, i.e., coordinates of the points of the search space
$\mathbf{x}^*$	the optimum location
$H(\mathbf{x})$	objective function
k	iteration number
$\mathbf{x}_k^i$	vector of coordinates of the i th particle at iteration k
$\mathbf{x}_k^*$	vector of coordinates of the best particle at iteration k
$\mathbf{x}_k$	general representation of a particle at iteration k
w_k^i	weight of the i th particle at iteration k
N_s	number of the particles
y_k	observation at iteration k
v_k	observation noise at iteration k
$H_{max,k}, H_{min,k}$	maximal and minimal objective function value belonging to the particles of iteration k
x_{max}, x_{min}	maximal and minimal boundaries of the decision variables
$\pi_k(\cdot)$	posterior pdf at iteration k
$\varphi_k(\cdot)$	pdf of the observation noise at iteration k
$p(\cdot)$	probability density
λ_k	parameter of the exponential version of $\varphi_k(\cdot)$
Δw	adjustable parameter representing weight ratio in the exponential version of $\varphi_k(\cdot)$
$\mathbf{u}_k$	a random variable drawn from the transition kernel at iteration k
α, β	parameters of the variance trajectory of the transition kernel
$E(\cdot)$	expected value
δ	tolerance
$\mathbf{c}_i$	vector that contains the centroids of the clusters
C_j	set of the particles belong to the j th cluster
$d(\cdot, \cdot)$	distance metric
K	number of the clusters
N_d	number of the dimensions
μ	mean
σ	standard deviation
M	number of iterations
$N_{particle}$	number of the particles within one cluster
$p(\cdot \cdot)$	conditional probability
$\mathbf{y}_{1:k}$	vector that collects the measurement data from y_1 to y_k
$Q^{(m)}$	cumulative sum of the weights of the first m particle

Acknowledgments

This research was funded by the ÚNKP-23-2 New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund. Besides, I owe thanks to my supervisor, János Abonyi, who allowed me complete freedom to follow my ideas and gave me maximal support to bring them alive.

Bibliography

- [1] Asegunloluwa Eunice Babalola, Bolanle Adefowoke Ojokoh, and Julius Beneoluchi Odili. A review of population-based optimization algorithms. In *2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICM-CECS)*, pages 1–7, 2020.
- [2] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.
- [3] J. Rawlings and D.Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing: Madison, WI, USA, 2009.
- [4] Enlu Zhou, Michael Fu, and Steven Marcus. A particle filtering framework for randomized optimization algorithms. pages 647–654, 12 2008.
- [5] Juan Camilo Medina and Alexandros A. Taflanidis. Adaptive importance sampling for optimization under uncertainty problems. *Computer Methods in Applied Mechanics and Engineering*, 279:133–162, 2014.
- [6] Xilu Wang and Yaochu Jin. Knowledge transfer based on particle filters for multi-objective optimization. *Mathematical and Computational Applications*, 28(1), 2023.
- [7] Bin Liu. Posterior exploration based sequential monte carlo for global optimization, 2016.
- [8] Yao Nan, Qin Jian-Hua, Zhu Xue-Qiong, and Wang Hong-Chang. Dynamic path planning based on improved particle filter optimisation for patrol robots. In *2022 7th Asia Conference on Power and Electrical Engineering (ACPEE)*, pages 1898–1903, 2022.
- [9] Adam Chehouri, Rafic Younes, Jihan Khoder, Jean Perron, and Adrian Ilinca. A selection process for genetic algorithm using clustering analysis. *Algorithms*, 10(4), 2017.

- [10] Shengxiang Yang and Changhe Li. A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments. *IEEE Transactions on Evolutionary Computation*, 14(6):959–974, 2010.
- [11] Bin Liu, Shi Cheng, and Yuhui Shi. Particle filter optimization: A brief introduction. pages 95–104, 06 2016.
- [12] Jos Elfring, Elena Torta, and René van de Molengraft. Particle filters: A hands-on tutorial. *Sensors*, 21(2), 2021.
- [13] Jiaqiao Hu, Yongqiang Wang, Enlu Zhou, Michael C. Fu, and Steven I. Marcus. *A Survey of Some Model-Based Methods for Global Optimization*, pages 157–179. Birkhäuser Boston, Boston, 2012.
- [14] Enlu Zhou, Michael Fu, and Steven Marcus. Particle filtering framework for a class of randomized optimization algorithms. *Automatic Control, IEEE Transactions on*, 59:1025–1030, 04 2014.
- [15] Shen Yin and Xiangping Zhu. Intelligent particle filter and its application to fault detection of nonlinear system. *IEEE Transactions on Industrial Electronics*, 62(6):3852–3861, 2015.
- [16] P.N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Pearson International Edition. Pearson Addison Wesley, 2006.
- [17] Laurence Morissette and Sylvain Chartier. The k-means clustering technique: General considerations and implementation in mathematica. *Tutorials in Quantitative Methods for Psychology*, 9:15–24, 02 2013.
- [18] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.