The handle http://hdl.handle.net/1887/22875 holds various files of this Leiden University dissertation

**Author:** Reehuis, Edgar  
**Title:** Guiding evolutionary search towards innovative solutions  
**Issue Date:** 2013-12-17
Integration of Exploration Criteria in the Search

In the previous chapter, criteria for inducing exploration in a search space were examined. In that first step, the resulting sampling behavior of selecting on such exploration criteria was studied, without including quality as primary objective. This chapter aims to conciliate quality, i.e., the exploitation criterion, and the exploration criterion in the search, minimizing conflicts between them and maximizing their combined performance, evaluated as the ability to find a varied set of high-performing solutions.

Basically, this means composing an appropriate selection scheme that takes both criteria into account. The exploration criterion has to be enabled to efficiently direct exploitation to new areas. The exploration criteria that are considered are dynamic, depending on the solutions encountered earlier during the search. This makes the search a sequential endeavor that requires, implicitly or explicitly, switching between phases of exploration and exploitation. While combinations of exploration values can be used (see, e.g., [Hester and Stone, 2012]), we look for a scheme combining a single exploration criterion and quality.

Niching

Niching methods (see, e.g., [Shir et al., 2010]) are selection schemes that do not apply an exploration criterion, in our sense of the term, but that are aimed at actively promoting diversity in their result set. A niching approach runs on top of a search algorithm and makes it target different optima in parallel, performing parallel optimizations in different niches in the search space. Niching is able to do so through assuming that

---

5
a certain minimum distance exists between the different optima. This niching radius plays a central role, a proper estimate of it is required for the process to work.

In [Shir et al., 2010], advanced niching approaches are described, intended specifically to run on top of the CMA-ES (see Chapter 2) and divided in fixed-radius and adaptive-radius schemes. Importantly, all these schemes are based on distance between solutions in the decision space, that is, based on comparing solution vectors directly. This allows to couple the development of the niching radius to the development of the search distribution in the CMA-ES, leading to adaptive-radius schemes, and to provide a general rule for initializing a fixed radius, but these options are not available in using an arbitrary distance measure.

As described in Chapter 3, a domain-specific distance measure will be used for valuing the diversity in found solutions, for evaluating the performance of a search method. If we want to use niching to directly promote diversity with respect to an arbitrary domain-specific distance measure, we are limited to fixed-radius niching. A fixed minimum distance has to be provided, quantified in domain-specific distance, that will be kept between different parallel optimizations. However, if sensibly can be assumed that distance in the decision space reflects domain-specific distance, the advanced niching approaches can be used as is to implicitly improve diversity with respect to the domain-specific distance measure.

Nevertheless, another aspect that further limits the applicability of niching is the need to start the search from a single fixed solution, further elaborated on in the following section. Typically, in using niching, the search is initialized widely in the search space, increasing the chance of identifying the requested number of niches from the start on. If less than the requested number of niches are available at any point during the optimization, niching explores for additional niches by reinitializing solutions, i.e., to solution vectors with random values. This exploratory behavior can be strengthened by using the dynamic exploration criteria studied in Chapter 4 to explore for new niches. In such a hybrid scheme, however, the added value of the parallel optimizations is doubtful, as is relied on sequential exploration to be able to start these optimizations. In this chapter we will therefore address inherently sequential schemes only for integrating the exploration criteria.
5.1 · Integration Schemes for Scenarios with Infeasibility

The aim of the search is to find a diverse set of high-quality solutions, but with a specific application in mind, in the domain of airfoil optimization, see Chapter 6. That application poses difficulties to reach this goal using multiple quality-based optimization runs (i.e., the default approach), as the search needs to be started from an initial solution that is known to be feasible. Feasibility of a solution is defined here as not violating structural constraints to such an extent that the solution’s quality cannot be approximated through aerodynamic simulation. The borders of feasible space are not known in advance, therefore, only a small search distribution around the initial solution is used at the start of the search. This makes an adaptive search method optimizing on quality more likely to converge to suboptimal local optima, close to the initial solution, and thereby different instances of the adaptive search more likely to produce similar results.

**Test Setup**

We capture some of these properties, namely, infeasible areas and a large amount of local optima, in the to-be-minimized composite test function $AckleyInfeasible$, see Figure 5.1, of which a two-dimensional instance is used with search domain $[-15,15]^n \subseteq \mathbb{R}^n$, $n = 2$,

$$
f_{\text{AckleyInfeasible}}(\mathbf{x}) = f_{\text{Ackley}}(\mathbf{x}) + \begin{cases} 
 c_1 \cdot f_{\text{Gaussians}}(\mathbf{x}) & \text{if } f_{\text{Gaussians}}(\mathbf{x}) > 0.25 \\
 0 & \text{otherwise} 
\end{cases}. 
$$  \quad (5.1)

$AckleyInfeasible$ is a combination of the Ackley function \cite{Ackley1987,Back1996}

\begin{align*}
    f_{\text{Ackley}}(\mathbf{x}) &= -c_1 \cdot \exp \left( -c_2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} \right) - \exp \left( \frac{1}{n} \sum_{i=1}^{n} \cos(c_3 \cdot x_i) \right) + c_1 + e 
\end{align*}

that with $c_1 = 20$, $c_2 = 0.2$, $c_3 = 2\pi$ has a surface covered with local minima in a decreasing trend towards the single global optimum of 0 at $(0,0)$, and the Gaussians function \cite{Graening2010,Reehuis2011}

$$
f_{\text{Gaussians}}(\mathbf{x}) = \max_{j \in \{1,\ldots,20\}} \left( \exp \left( -\frac{1}{2}(\mathbf{x} - \mu_j)'\Sigma^{-1}(\mathbf{x} - \mu_j) \right) \right) 
$$

that features 20 Gaussian kernels with centers $\mu_j$ and sharing covariance matrix $\Sigma = 1.6^2 \cdot I_n$ (i.e., variance times the $n$-by-$n$ identity matrix). The Gaussian kernels are used to introduce infeasible areas in the search space: We consider a solution to be infeasible
when it has a quality value higher than 22. The kernels have decreasing quality values from the center to the edge, allowing quality-based search to move back to space that is considered feasible. The positions of the kernel centers are chosen randomly in the subdomain \([-10, 10]^n\), but are the same for all function calls.

The search is always initialized in point \((10, 10)\), using a \((6, 12)\)-CMA-ES (see Chapter 2) as underlying search method with comma-selection and populations of 6 parents and 12 offspring\(^1\), and an initial stepsize \(\sigma_{\text{init}}\) of 0.3. For constraint handling, we add a variable penalty to the criterion that is selected on, i.e., a solution’s quality value and/or exploration value, that increases linearly with the extent of violation of the lower boundary of \(-15\) or the upper boundary of 15 per dimension.

Not all integration schemes that are described are considered for analysis of their induced sampling dynamics, only those that we deem promising are tested. Based on the outcome of visual analysis, the performance of certain schemes will be further analyzed using the quality/diversity-Pareto-front interpretation presented in Section 3.2.1, allowing for statistical comparison of attained hypervolume scores. As domain-specific distance measure for test problem AckleyInfeasible, the Euclidean distance between solution vectors is used.

Selected schemes are tested using the two promising exploration criteria found in Chapter 4. These are the distance-based novelty measure uniqueness \((Un)\) and

---

\(^1\) Larger population sizes are used than of the \((3, 6)\)-CMA-ES prescribed in Chapter 2 for two-dimensional search spaces, as this is required for the Pareto-selection scheme, see Section 5.2.
Figure 5.2 Exploration Scheme Required. The solutions generated by a single run on test problem AckleyInfeasible, of a (6,12)-CMA-ES and a (6,12)-IPOP-CMA-ES selecting on quality, are displayed (recently-generated solutions are dark-colored, early-generated solutions are bright-colored). The CMA-ES is run for 495 generations (6000 quality evaluations), and the IPOP-CMA-ES for 105 generations (23k quality evaluations, restarting and doubling the population sizes 5 times). For test function AckleyInfeasible, in combination with an initial stepsize of 0.3, it is clearly visible that an exploration scheme is required for reaching the global minimum at (0,0).

interestingness measure reducible error $(RE_{\text{mDP}})$. Interestingness measure $RE_{\text{mDP}}$ is based on memories of dispersion-in-prediction learning-based novelty scores. For Un and $RE_{\text{mDP}}$, the settings listed in Chapter 4 are used. When an integration scheme requires deviating from these settings, this is described in the section defining it. All methods and techniques used have been implemented using the Shark Machine Learning Library v2.3.43\footnote{http://shark-project.sourceforge.net}.

To illustrate the need for an exploration scheme, we run the CMA-ES and the IPOP-CMA-ES [Auger and Hansen, 2005] on AckleyInfeasible, see Figure 5.2, using the settings described above. IPOP is a restart scheme that reinitializes with increased population sizes upon meeting an convergence criterion: We use doubling of the population sizes and convergence determined as a stepsize smaller than $0.5 \times \sigma_{\text{init}}$. Both approaches show optimization to (repeatedly, in case of the IPOP-CMA-ES) converge to a local minimum close to the initial point $(10,10)$.

The remainder of this chapter is organized as follows: In Section 5.2, integration schemes that select on quality and exploration criterion in parallel are described. In Section 5.3, a different approach is used: Quality optimization and exploration are separated in distinct phases. Section 5.4 provides a summary of the chapter.
5.2 · Multiobjective Selection

A straightforward way of integrating an exploration criterion into quality-based search is taking both criteria into account in parallel. One has the choice between blending or aggregating both values into a single fitness value [Cuccu and Gomez, 2011, Cuccu et al., 2011a] and using a Pareto-based multiobjective approach that determines a ranking based on *non-dominance* with respect to both criteria [Mouret, 2011, Graening et al., 2010]. While being relatively easy to implement, the downside of these schemes is that both objectives can counteract each other.

As described in Chapter 4, our view on exploitation is that of greedy, local search. In exploiting the quality function, the search is therefore best guided by the quality function alone. In zooming in on a local optimum, the search densely samples an increasingly-small area of the search space. In doing so it is obviously hindered by a dynamic exploration criterion that continuously tries to move the search to less explored areas. Conversely, in exploration, the search is hindered by the quality objective in escaping from sufficiently-exploited optima, as their high quality works as an attractor. Nevertheless, as multiobjective schemes are often used, we discuss them here.

5.2.1 · Weighted Aggregation

Aggregating a quality value \( f(x) \) and exploration value \( EV(x) \) into a single fitness value is done based on a weighting parameter \( 0 \leq w \leq 1 \) [Cuccu and Gomez, 2011, Cuccu et al., 2011a],

\[
(w - 1) \cdot f(x) + w \cdot EV(x).
\] (5.4)

In order to make \( w \) of influence and mix both criteria meaningfully, both measures have to be normalized to values between 0 and 1 and transformed to be minimized [Cuccu and Gomez, 2011]. This can be done by determining the best and worst value (\( v_{\text{best}} \) and \( v_{\text{worst}} \)) occurring per criterion in the candidate solutions at each selection round (i.e., the offspring at each generation), and normalizing a value \( v \) as follows,

\[
v_{\text{norm}} = \frac{v - v_{\text{best}}}{v_{\text{worst}} - v_{\text{best}}}.
\] (5.5)

In addition to the hindering effect that both objectives have on each other, choosing an appropriate \( w \)-value is no easy task as the optimal setting will change during the search, depending on whether the method is zooming in on an optimum or should
be taken away from it after sufficient exploitation. Thus, ideally, the \( w \)-value is set dynamically, updated during the search based on detection of the end of either the exploitation or exploration phase. For instance, if the improvement of the average or best quality occurring in the population falls under a certain magnitude, \( w \) can be turned up to escape the current optimum, whereas \( w \) can be turned down if the best quality in the population shows improvement over a certain time window, in order to focus the search better [Cuccu and Gomez, 2011]. However, exactly how to adjust the \( w \)-value each time the search should switch between exploitation and exploration remains to be specified.

**Relation to Surrogate Utility Functions**

Certain utility functions used in surrogate-assisted optimization are actually implementations of weighted aggregation, but they combine approximated quality values \( \hat{f}(x) \) and associated variances \( s^2(x) \) instead of actual quality values and general exploration measures; examples are the lower confidence bound (LB) and expected improvement (ExI) functions, see Section 2.3.1. The central assumption in using these utility functions is that the actual quality value \( f(x) \) is a realization of a random variable, for instance, normally-distributed, with mean \( \hat{f}(x) \) and variance \( s^2(x) \),

\[
f(x) \sim \mathcal{N} \left( \hat{f}(x), s^2(x) \right).
\] (5.6)

An exploration measure is only sensibly applied in a surrogate utility function if it can be used to express the variance of a quality prediction, like in Equation 5.6. We are thus restricted to the model-derived exploration measures, i.e., learning-based novelty and interestingness, see Chapter 4. **Predictive variance** (PV) is by definition suitable, as it is taken equal to the variance \( s^2(x) \) that is readily available in certain surrogate modeling approaches, see Section 4.2.2.2. It is commonly used in ExI (e.g., [Jones et al., 1998, Emmerich et al., 2006]) and LB (e.g., [Emmerich et al., 2006, Lu et al., 2013]). **Prediction error** (PE) requires knowledge of the actual quality value so it does not make sense to apply it in ranking solutions based on predicted quality values. **Dispersion in predictions** (DP) can be used to derive the variance,

\[
s^2(x) \approx \left( \frac{0.68}{0.5} \cdot DP(x) \right)^2
\] (5.7)
assuming the predictions \( \hat{f}(x) \), see Section 4.2.2.3, to be distributed according to
\[
\mathcal{N} \left( \text{median} \left( \hat{f}(x) \right), s^2(x) \right).
\] (5.8)

DP is taken equal to the interquartile range between the predictions \( \hat{f}(x) \), and thus, as it spans two quartiles, represents 50% of the data in the distribution, whereas 68% of the data in the distribution lie within one standard deviation \( s(x) \) of the mean according to the 68–95–99.7 rule \cite{Tanton2005}. The mean is here median \( \left( \hat{f}(x) \right) \).

Interestingness expressed as reducible error is a decreased variant of the current error, a solution’s learning-based novelty value, and is obtained by subtracting the irreducible error that was determined in the region in which the solution lies, see Section 4.3.3.4. The interestingness measure \( \text{RE}_{m\text{DP}} \) can be used as estimate of the variance in place of DP,
\[
s^2(x) \approx \left( \frac{0.68}{0.5} \cdot \text{RE}_{m\text{DP}}(x) \right)^2.
\] (5.9)

\( \text{RE}_{m\text{DP}} \) decreases DP values and thereby the derived variance quantities for those solutions in areas that show little learning progress, leading to worse rankings for these solutions based on the utility functions.

5.2.2 · Pareto-based Selection

In Pareto-based selection, candidate solutions are ranked on both their quality value \( f(x) \) and exploration value \( \text{EV}(x) \), by iteratively determining fronts of non-dominated solutions and providing a secondary comparison criterion for solutions on the same front. Restricting the definition to two objectives, a candidate solution \( x \) is non-dominated if there is no other solution \( x' \) in the set of candidate solutions that is better than \( x \) with respect to both objectives simultaneously, or better with respect to one objective and at least equal with respect to the other objective.

Non-dominated sorting partitions the set of candidates into fronts based on domination, where the first front contains all solutions that are non-dominated by all other candidate solutions. The second front is obtained by excluding the solutions from the first front from the set, and then determining the non-dominated set under these conditions, and so forth. As non-dominated sorting results in solutions with the same rank, see Figure 5.3, a second criterion that discriminates between solutions with equal rank based on their placement on the mutual front is required. We use the crowding-distance criterion \cite{Deb2002}, aimed at evenly distributing points over the front.
5.2 - Multiobjective Selection

and Sprave [37] proposed the following preference relation to guide the search towards the (feasible) global optimal point:

Algorithm 2: Generation of a new individual

This will further be discussed in section IV.

Non-dominated sorting in (a) assigns each

candidate solution an index representing the front to which it belongs: The first front is non-dominated, and the later fronts are non-dominated if all lower-indexed points are removed. To discriminate between solutions on the same front, the crowding distance criterion in (b) first of all retains the extremal points, and then determines a ranking for the remaining points based on maximizing the distance between their direct neighbors on each side. Figures courtesy of [Emmerich et al., 2006].

Different authors have used this crowding-distance Pareto-based selection scheme for selecting on quality and exploration value in parallel. [Mouret, 2011] and [Lehman et al., 2013] use it to select on quality and distance-based novelty (sparseness), while in [Graening et al., 2010, Reehuis et al., 2011] it is used to select on quality and learning-based novelty (prediction error and dispersion in predictions). [Schaul et al., 2011] apply Pareto-based selection, but instead of using crowding distance, they randomly pick a single point from the first front. Furthermore, they use Pareto-based selection on expected improvement (ExI, see Section 2.3.1), which they derive from quality predictions and learning-based novelty expressed as predictive variance, in parallel to, again, learning-based novelty expressed as predictive variance. This hybrid multiobjective scheme — weighted aggregation and Pareto-based selection — seems unnecessarily complicated, although it might provide some benefit as via ExI, learning-based novelty is not directly involved in the ranking, but scales the distribution around the quality prediction used to calculate ExI.

Being a set-based selection scheme that promotes extremal solutions, using crowding-distance Pareto-based selection shows clear stagnation combined with the (3,6)-CMA-ES recommended for two-dimensional search spaces, see Chapter 2; in the experiments, we therefore use a (6,12)-CMA-ES. In Figure 5.4, the results of using this selection scheme together with the selected exploration measures Un and
Figure 5.4 Pareto-based Selection. The plots show the solutions generated by a single run on test problem AckleyInfeasible, of a (6,12)-CMA-ES selecting on quality, and Un (in (a) through (c)) or $\text{RE}_{m\text{DP}}$ (in (d) through (f)) using non-dominated sorting, at three time steps (recently-generated solutions are dark-colored, early-generated solutions are bright-colored). The top-row plots display the development of the Un scores (in (a) through (c)) and $\text{RE}_{m\text{DP}}$ scores (in (d) through (f)), where the red dots represent the lastly-selected parent population. In both setups, the CMA-ES is unable to target optima and results in sampling solutions from a search-space-wide, line-shaped distribution.
RE_mDP, see Section 5.1, are displayed. While being elegant in the sense of omitting weight parameters, the applied Pareto-based selection prevents the CMA-ES from targeting optima and for both exploration measures makes that it samples points from a line-shaped search distribution extending between the edges of the search space.

### 5.3 · Separate Exploration Phases

While exploring, the search should generally diverge, but in exploiting it should converge. Therefore, instead of dynamically combining quality-optimization and exploration objectives, we follow [Cuccu et al., 2011a] and strictly separate exploration from exploitation. The idea is to use a separate exploration phase to find a new starting position for the quality-based optimization. In the exploration phase, ranking solutions is done on exploration value, and in the optimization phase, selection is on quality only.

This separation requires switching between phases and thus a way of deciding when to end either the exploitation phase or the exploration phase, expressed as stopping conditions for the search algorithm. Through its convergent nature, for the exploitation phase stopping conditions are available. A list of possible stopping conditions for the CMA-ES optimizing on quality is given in [Auger and Hansen, 2005]. We follow [Cuccu et al., 2011a] and use a stopping condition based on the stepsize of the search distribution. For the exploration phase, being divergent, it is less straightforward to formulate a stopping condition. An idea is looking at the development of the solutions’ quality: After deteriorating, improving quality values can indicate that a “ridge” has been crossed, potentially entering a new optimum’s basin of attraction. How exactly to determine the trend in the development of quality values, for instance the size of the time window over which to look, has to be thought of.

For the application at hand, however, involving infeasible regions, we can formulate a stopping condition based on feasibility. Arrival in infeasible space will be used as stopping condition, assuming that this approach is sufficient for leaving basins of attraction of earlier-converged-to optima. Intuitively, one might suggest passing through the infeasible region entirely before switching to quality-based optimization. Yet, for the actual problem in the domain of aerodynamic optimization of flow bodies, it is uncertain whether a feasible region will lie beyond it, see Chapter 6. A variable penalty is therefore included in the quality value based on the amount of infeasibility,
allowing the quality-based search to return to feasible space.

Because of the difficulty in coining a general stopping condition for the exploration phase, [Cuccu et al., 2011a] opt to not actually perform an exploration phase but instead select the solution, from all points already visited, that scores best on the used dynamic exploration measure at that moment in time. As in our problem setup we start from a fixed initial solution, instead of an initial population distributed widely over the search space, we have to use an active exploration phase, as in this scenario all solutions that have been visited up to a certain point can reasonably be stated to be in the basin of attraction of an earlier-converged-to optimum. By active we mean that new, unseen solutions are sampled during exploration. This way, there is the possibility of skipping over high-quality regions, as we do not select on quality, brought to bear by [Cuccu et al., 2011a] as well in motivating their passive exploration phase. However, in our approach, laid out below, the regions traversed in the exploration phases are not prevented to be visited in a later quality optimization or exploration phase.

Thus, checking feasibility of solutions allows for a convenient stopping condition for the exploration phase. It does, however, require evaluating solutions on quality in order to determine their feasibility. Strictly speaking, the exploration phase can be stated to “waste” quality evaluations, as it does use quality function evaluations but only for checking feasibility, not for driving the search directly.

5.3.1 · Alternating Restart Scheme

As presented in [Reehuis et al., 2013b, Reehuis et al., 2013c], we employ a restart scheme that alternates between phases of exploration and quality optimization, and starts with a quality-optimization phase. The scheme is summarized in Technical Note 5.1.

At the start of both the exploration and the quality-optimization phase, the underlying search method has to be reinitialized. A different or differently-configured search algorithm can be used per phase, but we employ the same CMA-ES for both. Reinitializing means for the CMA-ES that all variables involved in shaping its search distribution are reset, including the stepsize which is reinstated to its, problem-specific, initial value $\sigma_{\text{init}}$.

As stated before, each phase requires a stopping condition to be formulated for the search method. The quality-optimization phase will be stopped when the CMA-ES’
Technical Note 5.1 Alternating Restart Scheme for the CMA-ES

1) **Quality-optimization phase:**
   - Initialize stepsize at $\sigma_{\text{init}}$;
   - Rank solutions on quality value;
   - **Stopping condition:** Stepsize smaller than $0.5 \times \sigma_{\text{init}}$.

   Take the best-ranked solution of the current generation and
   - add it to the archive of found optima;
   - start the exploration phase from this solution.

2) **Exploration phase:**
   - Initialize stepsize at $\sigma_{\text{init}}$;
   - Rank solutions on exploration value, using
     - Un, reference set: All solutions generated in quality optimization;
     - RE$_{\text{mDP}}$, reference set: All solutions evaluated on quality.
   - **Stopping condition:** Stepsize smaller than $0.5 \times \sigma_{\text{init}}$, or the best-ranked solution of the current generation
     1. exceeds a *hard*-limit quality value;
     2. exceeds a *soft*-limit quality value for a number of generations;
     3. violates other domain-specific feasibility conditions.

   Start the quality-optimization phase from the solution
   - with the highest-found Un score in the current exploration phase, or
   - best-ranked on RE$_{\text{mDP}}$ in the current generation.

**Quit** the restart scheme when all available quality evaluations have been used, at any point during execution.
stepsizes becomes smaller than $0.5 \times \sigma_{\text{init}}$, considering the search to have converged to an optimum. The exploration phase is also stopped when the stepsize is smaller than $0.5 \times \sigma_{\text{init}}$, to account for possible stagnation, and is stopped when infeasible space is entered, considered such when the best-ranked solution with respect to the exploration measure:

1. Exceeds a **hard**-limit quality value — for AckleyInfeasible, higher than 30;

2. Exceeds a **soft**-limit quality value for a number of consecutive generations — for AckleyInfeasible, higher than 22 for 5 consecutive generations;

3. Violates other domain-specific feasibility conditions — these are not present in AckleyInfeasible.

When applying Un in the exploration phases, search is performed on a *static* landscape as the used reference set consists of the solutions generated during the quality-optimization phases. The solutions that are selected in the exploration phase, i.e., the parent individuals, are evaluated on quality for feasibility checking, but not added to the reference set. Using Un, we aim to put maximum distance to solutions in the regions of attraction of the converged-to optima.

When applying RE$_{m\text{DP}}$ in the exploration phases, search is performed on a *dynamic* landscape: The reference set for training the model contains all solutions evaluated on quality, so including the solutions that were evaluated in the exploration phases for feasibility checking. This is required because the maximum of the interestingness landscape is close to the lastly-sampled solutions. We have to keep adding solutions to the reference set to make this maximum and thereby the search move away from where exploration was started. Per generation, the generated offspring serve as test set for determining the modeling errors using DP, see Section 4.3.

The alternating restart scheme is only allowed to switch between phases if in the last phase, exploration and quality-optimization phases alike, at least one feasible solution was generated with respect to conditions 1. and 3. listed above. If not, instead of switching phases, the last type of phase is started again, from the next restart point in line. For an exploration phase, this means that not the best-ranked solution from the last generation of the previous quality-optimization phase is taken, but the runner-up solution, continuing in that fashion and ending-up in a before-last generation if need be, after multiple attempted reinitializations. For a quality-optimization phase
this means, in case of Un, that the runner-up solution of the previous exploration phase is used, and so forth, and, in case of \( \text{RE}_{\text{mDP}} \), that the runner-up solution in the last generation of the previous exploration phase is used, moving back into earlier generations if necessary, like described above.

Using separate exploration and exploitation phases omits the need to perform a clustering step to isolate found optima: The best solution of the last generation of a quality-optimization phase is stored as approximated optimum. Through the exploration phases, the distance between the found optima, with respect to the used domain-specific distance measure, is promoted, either directly in Un, or implicitly in \( \text{RE}_{\text{mDP}} \) by aiming for maximum learning progress.

**Level-set Approximation**

A level set contains all solutions in the search space with quality equal to or better than a given threshold quality value, see Section 3.2.1. Level-set approximation aims to represent the level set using a limited number of solutions, which are therefore to be spread optimally over the area in which solutions satisfying the level-set constraint lie. The distance between solutions is determined using the generally-applicable Euclidean distance between the solution vectors, or a different domain-specific distance measure. A level-set approximation is valued by the diversity in the set of solutions that make up the approximation, which is aimed to be maximal according to the used set diversity measure. This set diversity measure applies the domain-specific distance measure to pairs of solutions in the set.

One successive exploration and exploitation phase can combined be seen as a variation operator in approximating a level set, delivering a found optimum as a new solution that is to be considered for the approximation. Such a variation operator would be applicable in a scheme such as *ELSA* [Emmerich et al., 2012, Emmerich et al., 2013] that aims to approximate a level set with a given threshold \( f_{\text{thres}} \) using a fixed-size population. However, considering the computational cost involved in applying this “variation operator”, one might want to keep all generated candidate solutions and in a post-processing step determine to adopt a certain \( f_{\text{thres}} \) based on the solutions generated.

This is exactly what is done in running the alternating restart scheme and afterwards determining the diversity of all possible level sets in its result set, i.e., the set of found optima, as laid out in Section 3.2.1. That way, no pre-defined \( f_{\text{thres}} \) is used,
and although one does not optimize on diversity, the distance between found optima is promoted by exploring for restart points that lie far away from earlier-generated solutions. Nevertheless, should one have a pre-defined $f_{\text{thres}}$ in mind, it can be used as additional criterion for stopping the quality-optimization phases. In the current setup, governed by a minimal stepsize to be reached, a quality-optimization phase will only stop when a solution has been converged to that has a sufficient level of local optimality. Technically, however, within a level-set approximation, local optimality of the constituents is not a requirement, only satisfaction of the level-set constraint.

**Experiments**

We apply the alternating restart scheme on test function AckleyInfeasible, running on top of a CMA-ES, with settings as described in Section 5.1. Within the exploration phases, we test Un, as defined and with settings described in Section 4.2.1.2, and RE$_{\text{mDP}}$, as defined and with settings described in Section 4.2.2.3 (training data of $\gamma = 5$ successive generations of evaluated solutions per local model stack). A third, simplified approach is included, *line explore*, inspired by the dynamics of the RE$_{\text{mDP}}$ exploration phases. It features a similar division in quality-optimization and exploration phases, but exploration is performed by just sampling a random vector from a multivariate normal distribution, $\mathcal{N}(0, \sigma^2 I_n)$ with $\sigma = 0.3$ (equal to $\sigma_{\text{init}}$ for AckleyInfeasible), and iteratively adding it to the start solution, resulting in moving in a straight line through the search space. After each addition, the resulting solution is evaluated on quality to determine whether one of the infeasibility stopping conditions was met.

Plots of all solutions generated in a single run by each of the three methods, the alternating restart scheme using Un, the alternating restart scheme using RE$_{\text{mDP}}$, and the alternating restart scheme using line explore, are shown in Figure 5.5 through 5.7. Of these three single runs, the results are summarized in Figure 5.8 using the quality/diversity-Pareto interpretation described in Section 3.2.1. Furthermore, quantifying the surface dominated in quality/diversity space as a hypervolume score, the results of 100 runs per method are compared in Figure 5.9.

Looking at Figure 5.5, after quick convergence to a local optimum close to the initial solution of $(10, 10)$ (equal to the behavior of the unassisted CMA-ES, see Figure 5.2), an exploration phase is started in the direction of the center of the search space, where the Un landscape shows higher values. With exploration quickly ending up in
5.3 Separate Exploration Phases

Figure 5.5 Restarting — Un. The solutions generated by a single run on test problem AckleyInfeasible, of the restart scheme based on a (6, 12)-CMA-ES alternating between optimization on quality and exploration using uniqueness, at three time steps (recently-generated solutions are dark-colored, early-generated solutions are bright-colored). The middle-row plots show the solutions generated in the exploration phases and the bottom-row plots the solutions generated in the quality-optimization phases. The top-row plots display the development of the Un scores, where the red dots represent the last-selected parent population in case of quality optimization and the yellow squares in case of exploration. The alternating restart scheme is able to leave the lower-right corner of the search space, and is able to locate the global optimum at (0, 0), as well as multiple alternative optima.

Infeasible space, a few successive exploration phases are required for exploitation to leave the lower-right corner of the search space. In generation 122, exploitation is restarted in point (9.7, 5.2), after which it is able to autonomously locate the global optimum. The alternating restart scheme continues, finally delivering a result set
Figure 5.6 Restarting — $\text{RE}_{mDP}$. The solutions generated by a single run on test problem AckleyInfeasible, of the restart scheme based on a (6,12)-CMA-ES alternating between optimization on quality and exploration using $\text{RE}_{mDP}$, at three time steps (recently-generated solutions are dark-colored, early-generated solutions are bright-colored). The middle-row plots show the solutions generated in the exploration phases and the bottom-row plots the solutions generated in the quality-optimization phases. The top-row plots display the development of the $\text{RE}_{mDP}$ scores, where the red dots represent the lastly-selected parent population in case of quality optimization and the yellow squares in case of exploration. Exploration using $\text{RE}_{mDP}$ constrains the width of the search distribution, showing relatively-slow but directed movement through the search space, and allowing multiple optima, including the global optimum at (0,0), to be located.

containing 20 (local) optima in using 6000 quality evaluations.

Whereas exploration on Un increases the width of its search distribution after being initialized, exploration on $\text{RE}_{mDP}$, see Figure 5.6, shows the familiar interestingness peak that moves through the search space and that keeps the width of the search
5.3 Separate Exploration Phases

Figure 5.7 Restarting — Line Explore. The solutions generated by a single run on test problem AckleyInfeasible, of the restart scheme based on a (6,12)-CMA-ES alternating between optimization on quality and line exploring, at three time steps (recently-generated solutions are dark-colored, early-generated solutions are bright-colored). Following a straight line for finding restart points for the exploitation phases is a relatively simple approach that allows for locating multiple optima, including the global optimum at (0,0).

distribution at moderate levels. From the plots of generation 805, it is clear to see that exploration brings quality-optimization to new areas to exploit. In doing so, it moves through the search space at a lower speed than exploration on Un, and as such delivers a result set containing fewer optima, 8, after using 6000 quality evaluations.

Different runs of the alternating restart scheme reach a different number of generations. This is because the exploration phases do not require evaluating all solutions but only the selected parents, thereby allowing for more generations in total. For the restart scheme using line explore, see Figure 5.7, the exploration phases are not counted in the number of generations, as only one solution is generated and evaluated per iteration. As can be seen, line explore is a simple approach that allows the CMA-ES to leave the region of attraction of the optimum found by default, to locate the global optimum, and produces 15 alternative optima in using 6000 quality evaluations.

From Figure 5.8, it is visible that with 20 approximated optima, Un generated more points than the other methods, resulting in the highest hypervolume score of 63.2. Line explore delivered only a few optima less, but the majority of these were approximations of the global optimum (11 of 15 found in total). Similar solutions do not contribute much to the surface dominated in quality/diversity space, and hence its hypervolume score of 37.8 is the lowest of the three runs. RE\textsubscript{mDP}, with its slower-moving exploratory behavior, delivers only 8 approximations of optima, but 6 of these are sufficiently distinct to result in a hypervolume score of 39.6.
Figure 5.8 Single-run Performance Comparison. Per single run on AckleyInfeasible, the quality threshold and diversity score is plotted for each possible level set in the result set. Interpreting these pairs jointly as a Pareto-front approximation, we visualize the dominated surface in quality/diversity space. Based on a reference point of (20,0), uniqueness has an obtained hypervolume of 63.2 (result set of 20 solutions), reducible error of 39.6 (result set of 8 solutions), and line explore of 37.8 (result set of 15 solutions).

Finally, based on statistical comparison of attained hypervolume scores, see Figure 5.9, Un clearly outperforms RE_{mDP}, while line explore ends up in between. Nonetheless, looking at hypervolume scores only does not show the complete picture. The scores are greatly influenced by the number of solutions in a result set, through the used set diversity measure $D_{SP}$ that maximally returns a score equal to this number (see Section 3.2.1), and through the fact that larger sets are likely to dominate a larger surface in quality/diversity space. We therefore include a statistical comparison of the number of solutions found per method, showing a similar spread over the three methods as for the hypervolume scores. When we then, per run, divide the hypervolume score by the number of solutions found, Un and line explore show similar hypervolume per solution, but RE_{mDP} now has a higher median outcome. It should be noted here that the RE_{mDP} hypervolume-per-solution scores show much more variation though, in positive and negative directions. Lastly, another performance aspect considered is the quality value of the best solution found, which is to be minimized. Here Un and line explore again show similar performance, with RE_{mDP} falling a bit behind.
5.4 · Summary

When integrating dynamic exploration criteria into quality-based search, one has the choice between selecting on quality and exploration criterion in parallel, or dividing the search into distinct exploitation and exploration phases. Although the first approach is easier to implement, not requiring conditions for switching phases and performing reinitialization at the start of a new phase, it is noted that exploitation and exploration induce inherently conflicting dynamics. Increasingly local search is performed in exploiting, while exploration continuously drives the search away from areas that were visited already. Separation in dedicated phases therefore constitutes the more efficient approach, preventing that the objectives can hinder each other.

A certain level of to-be-attained accuracy, expressed as the stepsize of the search distribution, can intuitively be used as stopping for the exploitation phase. On the other hand, because of its divergent nature, formulating a general stopping condition for the exploration phase is less straightforward. The real-world application to which the developed methods are to be applied, however, includes regions with intolerable

![Figure 5.9 Statistical Comparison.](image)

Per method, of 100 runs on test function AckleyInfeasible, the hypervolume score of each result set in quality/diversity space is calculated, displayed in (a). The hypervolume score is influenced by the number of solutions per result set, displayed in (c). Hypervolume score divided by the number of solutions in the result set is therefore displayed in (b). Lastly, another performance aspect considered is the to-be-minimized quality value of the best solution found per run, shown in (d). Un produces the largest result sets and this helps to attain the greatest hypervolume scores of the three methods. When hypervolume is expressed per solution, it is clear that despite its lower total hypervolume scores, RE$_{mDP}$ is still a method to be considered for the actual real-world application.
solutions in its search space. After converging in the exploitation phase, we can start exploring until we end up in such an infeasible region, under the assumption that this puts enough distance to the region of attraction of the earlier converged-to optimum. Quality-based search is then restarted from this point close to the boundary of infeasible space, including a variable penalty to be able to move back to a feasible solution.

This results in what is termed the alternating restart scheme, here, but not specifically, run on top of the CMA-ES, and tested in employing the two promising exploration measures resulting from Chapter 4: Distance-based novelty measure uniqueness (Un) and interestingness measure reducible error, based on memories of dispersion-in-prediction novelty (RE$_{mDP}$). Furthermore, a third baseline method of exploring in a random straight line was included.

On the two-dimensional instance of test problem AckleyInfeasible, Un performs best with respect to hypervolume dominated in quality/diversity space, an approach laid out in Chapter 3 for measuring performance with respect to the aim of finding diverse high-quality solutions. RE$_{mDP}$ comes in last, and also has the smallest number of solutions found per run. When accounting for this, RE$_{mDP}$ does show the highest hypervolume-per-solution scores.

RE$_{mDP}$-induced exploration is slow in moving through the search space compared to the other two methods. This results in smaller result sets, and lower hypervolume scores. The actual application in optimization of airfoils that we have in mind for the developed methods involves a higher-dimensional search space. Moreover, Ackley-Infeasible might not sufficiently capture the essence of this application that includes more infeasibility characteristics. In the following chapter, we will therefore employ the alternating restart scheme with each of the three exploration approaches on the actual real-world problem.