Author: Reehuis, Edgar
Title: Guiding evolutionary search towards innovative solutions
Issue Date: 2013-12-17

7

Conclusions and Outlook

The main goal of this work is to develop a method that, operating on top of an *Evolutionary Algorithm*, increases the likeliness of finding *innovative solutions*. This likeliness is laid out to be increased with the *diversity* of the solutions found, provided that they are of sufficient quality. The developed method needs to be applicable in a scenario in which the search is required to be started from a single, fixed solution. Therefore, a scheme is envisioned in which the search is performed in a *sequential* fashion, zooming in on a locally-optimal solution, and then exploring for a new potentially high-quality region based on a *memory* of solutions encountered earlier in the search. Two *exploration criteria*, one using an *archive* of earlier solutions as memory and the other deriving from a *surrogate model* trained on earlier solutions, were established to be worthwhile for integration into quality-based search. The resulting schemes were applied to a real-world airfoil optimization task, showing both to perform better than the baseline method of multiple standard optimization runs. The model-based approach delivers the best results, in the sense that it finds more solutions, more diverse solutions, and better-quality solutions than the baseline method.

Next, in Section 7.1, the thesis is summarized in more detail with the most-important conclusions listed, while Section 7.2 provides future research directions.

7.1 · Summary

An innovative solution is a *product design* that has the potential of leading to an *innovation*, that is, the resulting innovative product getting adopted by the end users. *Innovativeness*, the property of being innovative, is defined in this work as being *novel* and of *high-quality*, both with respect to the comprehensive *reference set* of *state-of-the-art* solutions in the application domain. The state-of-the-art solutions are the established highest-quality solutions, generally regarded as such by the engineering community. Novelty is the distance to the closest solution from the used reference set, with respect to a *domain-specific distance measure*. This domain-specific distance measure is used to isolate and compare on those solution aspects that are relevant to the application.

Automated determination of a solution's true innovative value is complicated because of the need of compiling the comprehensive set of state-of-the-art solutions. On the one hand, it is difficult to take all established high-quality solutions into account, on the other hand, it is not easy to formulate patented solutions in such a way that they can be interpreted and compared to automatically. Instead of searching for innovative solutions directly, we therefore aim to deliver *diverse*, high-quality solutions, where the diversity of a set of solutions is evaluated with respect to the same domain-specific distance measure as is used in determining innovativeness. The assumption is that with increased diversity and quality in a result set, the chance for actual innovative solutions to be present in it increases as well.

The envisioned approach for finding diverse high-quality solutions is through incorporating an exploration criterion into quality-based search, and making the search alternate between *exploitation* (i.e., optimization on quality) and exploration in a sequential pattern. To steer clear of areas that were visited already, exploration needs a memory of where the search was before. As memory, an archive of earlier encountered solutions can be used, or a surrogate model that is trained on the earlier encountered solutions. The resulting *online* novelty of a solution, for usage as exploration criterion, is then either the *distance* to the solutions in the archive (*distance-based novelty*) or the *error* that the surrogate model makes in its *prediction* for it (*learning-based novelty*). The surrogate model approximates the mapping from solution to quality value.

In applying distance-based novelty, it is straightforward that diversity with respect to the same domain-specific distance measure is promoted. In selecting on learning-based novelty, on the other hand, the assumption is that solutions with maximum error improve the surrogate model most upon including them in model training: Learning-based novelty is a predictor for the *learning progress* that a solution leads to. Potentially, by following a pattern of exploration that maximizes learning progress, solutions are generated that are distant from each other with respect to the domain-specific distance measure. Among others depending on the used domain-specific distance measure, possibly, exploration moves through the search space more efficiently in maximizing learning progress than in selecting on distance-based novelty.

Testing on an artificial function reveals that the simplified distance-based novelty expression that was put forward (*uniqueness*) does not perform worse than the original formulation (*sparseness*): Uniqueness (Un) is the distance to the closest solution in an archive of all generated solutions. Learning-based novelty, on the other hand, turns out to be insufficient as predictor of learning progress: It needs to be extended by an approach that accounts for areas in which the model does not improve, to prevent exploration from stagnating, that is, getting stuck in such an area. The composite approach is termed the *interestingness* of a solution, deriving from learning-based novelty but providing a better prediction of the learning progress based on the *earlier-observed* modeling errors in the *region* in which a solution lies.

Of four tested, *reducible error* (RE) is the best-performing interestingness expression: It substracts an average of earlier-observed errors from the average of recent errors in a certain region, and thereby penalizes regions in which the error stays high. Herein, the assumption is preserved that high (recent) errors lead to high learning progress. For determining the modeling errors, *dispersion in predictions* (DP) is used, a learning-based novelty expression that *estimates* the modeling error by comparing the predictions made by multiple surrogate models. The resulting measure is denoted as RE_{mDP}, indicating that the observed errors in a region are averaged as a *long-term* and a *short-term memory*, obtained through exponential smoothing at different rates.

Novelty and interestingness express deviation from available knowledge, not the chance for high quality. This is in line with the view of exploration and exploitation having inherently conflicting dynamics: While exploring, the search *diverges*, and in exploiting, it *converges*. In including an exploration criterion in quality-based search, we therefore strictly separate exploration from exploitation. Quality-based optimization is run until convergence, i.e., the search distribution from which new points are generated has reached an indicated minimum magnitude, after which we explore for a new starting point for quality-based optimization. Through their divergent nature, it is not straightforward to formulate a similar *stopping condition* for the exploration phases.

However, the real-world application on which the developed methods are intended to be applied includes large regions with *intolerable* solutions. Of these "*infeasible* regions", the boundaries are unknown in advance. Therefore, the search is required to be started from a basic solution that is known to be feasible. On the other hand, we *can* conveniently use entering infeasible space as a stopping condition for an exploration phase, in which the assumption is that this is sufficient for leaving the *basin of attraction* of the *optimum* that quality optimization converged to prior to start of the exploration phase.

The intended real-world application involves the optimization of a *stator blade* design in a small *turbofan* engine that is intended for propelling small business jet aircraft. In a *gas turbine*, non-moving stator blades are installed after the moving *rotor blades* to straighten the air flow. The quality of a solution, which is represented as a vector containing 32 variables, is approximated using *aerodynamic simulation*. As such, establishing this quality value is computationally intensive. After defining a domain-specific distance measure, deriving from the two-dimensional stator-blade *profiles* that are obtained by decoding the vector representation, we apply distance-based novelty expression Un and interestingness expression $\mathrm{RE}_{\mathrm{mDP}}$ using the alternating quality-optimization/exploration scheme described above. Next to usage in Un, the domain-specific distance measure is required for determining the diversity of the produced result sets, to compare performance between methods.

As baseline method, multiple standard quality-based optimization runs are used. Five of such runs account for the same quality-evaluation budget as is used for an exploration-assisted run, and can hence be seen as a single run that has been *restarted* four times. Assisted by interestingness-based exploration, the *CMA-ES*, an Evolutionary Algorithm with advanced online adaptation of its search distribution, delivers more solutions, more diverse solutions, *and* solutions of greater quality than five of the unassisted CMA-ES runs combined. Exploration based on Un provides more diverse and higher-quality solutions than standard optimization as well, but induces slower-moving exploration than $\mathrm{RE}_{\mathrm{mDP}}$ and thereby results in less solutions found, which influences the diversity scores and, implicitly, the best quality value found. As was surmised, in exploration based on learning progress, it can occur that the underlying optimizer, i.e., here the CMA-ES, is presented with a search landscape that it can more-efficiently traverse than the landscape derived from the domain-specific distance between solutions.

7.2 · Outlook

The interestingness-based exploration variant, though providing the best results, is an involved method, relying on multiple layers of components. Implementing it is complicated, with the risk of easily making mistakes in the process. It should be examined which parts are essential in its performance, and whether a similar induced pattern of exploration can be attained using a less complex setup. This examination was started using the *line-explore* method, which simply explores in a randomly-chosen straight line through the search space, but which was clearly outperformed by both Un and $\mathrm{RE}_{\mathrm{mDP}}$. However, in this setup, the gradient for the line was chosen from an *isotropic* probability distribution, parameterized-equally in all dimensions, which can possibly be done more cleverly by taking the adapted search distribution of the CMA-ES into account. When considering to apply the methods presented in this thesis, it is recommend to start with the less-performing, but simpler to implement archive-based exploration variant Un. Nevertheless, despite being more complex, on the real-world application, $\mathrm{RE}_{\mathrm{mDP}}$ *is* computationally less expensive than the Un scheme, as the latter requires comparing to all earlier generated solutions using the domain-specific distance measure.

Furthermore, the integration scheme by which both exploration variants are introduced into quality-based search depends on regions of intolerable solutions to exist in the search space, serving as stopping condition for the exploration when such an area is entered. To make the developed measures applicable to problems without similar infeasible regions, alternative stopping conditions are to be examined, for instance based on the development of the quality values encountered while exploring.

Lastly, in the real-world airfoil application, domain-specific distance is ideally derived from the *flow-field* approximations that are obtained through simulation and from which the quality values are determined. Ways of using these approximations, which are available as two-dimensional color plots, to compare solutions are to be studied, for instance by *morphing* the blade profile in one such image into a profile in another image, and then determining the difference between the resulting images pixel-wisely.