Adaptation of Optimization Algorithms to Problem Domains by Transfer Learning

Chris Reinke, Kenji Doya
Okinawa Institute of Science and Technology
Onna-son, Japan
chris.reinke@oist.jp

***Abstract*—Optimization is one of the most important problems in science and engineering. Common optimization algorithms are designed to work for a large set of problems, but not necessarily to be efficient for specific domains. We propose a new transfer learning approach to adapt optimization algorithms to specific problem domains. Our approach analyzes solved problems of a domain to identify areas in the search space where good solutions are expected for this domain. Knowledge of these areas is used to improve the optimization algorithm performance of unseen problems of the same domain. Because of its general design, our method can be applied to a wide range of problems and algorithms.**

Keywords—optimization, transfer learning, CMA-ES, Powell's Conjugate Gradient

I. Introduction

Optimization algorithms are designed to maximize (or minimize) parameters $\theta=(\theta_1,\theta_2,\ldots,\theta_K)\in\Theta$ of a real-valued function $f(\theta)\in R$. The goal is to identify the global optimum $\theta^*=argmax_\theta f(\theta)$, but functions have often several local optima θ' that have the optimal function value only in their neighboring area. Many optimization algorithms start from a point θ^{start} in the parameter space and evaluate the function for different parameters near this point to identify a nearby local optimum (Fig 1, b). To find the global or a good local optimum the optimization should be started from several points distributed over the whole parameter space Θ.

Optimization with common algorithms such as CMA-ES [3] or Powell's Conjugate Gradient [4] is designed to handle a wide range of problems by searching the whole parameter space Θ. This requires many function evaluations and is problematic if evaluations are time intensive. We propose a transfer learning approach to improve the performance of optimization algorithms by adapting them to the problem domain to which they are applied. In transfer learning, knowledge from solved source problems $F^S = (f^S_1, \ldots, f^S_N)$ is used to improve optimization performance on a related target problem f^T. Our approach identifies from the sources F^S start points θ^{start}, that lead the optimization on the target f^T toward good local optima with few function evaluations. Furthermore, in the case of a large or an infinite number of possible source functions, we introduce a procedure to select a subset of these functions which provide the important information to improve optimization on the target f^T.

A possible application scenario is the optimization of walking behavior for a robot. The walking controller has several parameters and is evaluated by the speed of the robot. Evaluating parameters on the robot is time intensive, because it has to move for several seconds. To solve this problem the controller could be optimized on a physical computer simulation, which is faster to evaluate. Nonetheless, a simulator can never accurately replicate the actual robot. Parameters for the walking controller that work in a simulator usually do not work well on the actual robot. In our approach, several simulators that differ, for example, in their friction coefficients, would be used as sources F^S. The optimization problem of the walking controller differs between each simulator and the actual robot, but they are still part of the same problem domain. The problem would be analyzed on each simulator to identify regions in the parameter space of the walking controller that have optima with a high performance. The acquired knowledge would then be used to reduce the number of function evaluations of the optimization on the target task f^T, the actual robot.

II. Method

Our approach identifies a solution space model $M=(a_i,\ldots,a_K)$ that describes K areas in the parameter space Θ of a problem domain (Fig. 1, c) with $a_i = (y_i, \theta_i, E_i)$. Each area has local optima (solutions) with similar function values. Areas are described by the mean function value y_i of their local optima, the mean position θ_i of their optima in the parameter

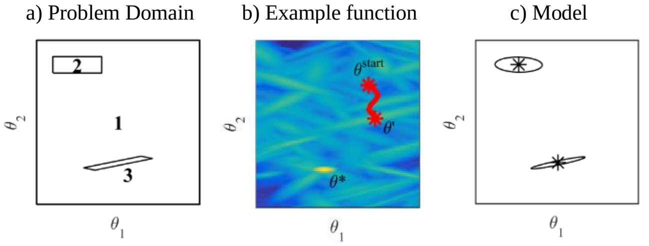

Fig. 1. (a) A 2-dimensional problem domain. Area 1 holds many inferior local optima with values around 0.6. Either area 2 or 3 holds the global optimum with a value around 1. (b) A function sampled from the problem domain with the search patch (red) of an optimization algorithm that starts at point θ^{start} and identifies a nearby local optimum θ'. The color indiciates the function value. The global optimum θ^* is located in area 3. (c) The identified model after analyzing 10 source functions from the problem domain. The two areas that hold the global optimum have been correctly identified. Their mean position (point) and their extend (ellipsis) are shown.

space and the extent of the area as an ellipsoid E_i in the parameter space.

The model is generated by analyzing the source problems F^S. For each source, several local optima are identified by using standard optimizations starting from several locations θ^{start} in the parameter space. Afterward, optima with nearby locations in the parameter space and similar function values from all source problems F^S are clustered together with the OPTICS algorithm [1]. The clustering is density based and uses the locations of the optima. The ranges to decide which values are regarded as similar are given by the user. Each cluster is then represented as an area in the model. The mean value y_i, the mean position θ_i, and the extent E_i are computed based on all optima from all sources F^S that are part of the cluster.

The model is then used to improve the optimization efficiency on the target task f^T by focusing the search on areas in the parameter space where optima with high function values are expected. The procedure is composed of successive local optimizations. For each optimization, an area a_i from the model is selected according to its mean function value y_i. Areas with high values have a higher probability to be selected. A point inside the area is sampled as the starting point θ^{start} for the optimization. The model is also used to adapt meta parameters such as the step length of the local optimization algorithm for its search in the area.

III. Experiments

We tested our method on functions generated by a modified Max Set of Gaussian Landscape Generator [2]. The generator samples functions for problem domains that are described by areas (Fig. 1, a). Each area defines a distribution over its function landscape, i.e. it holds distributions about the form, the number, and the function values of its local optima. Based on this description arbitrary many functions can be sampled (Fig. 1, b) for a problem domain.

For each experiment 10 source functions F^S were used to identify a model about their problem domain. Because arbitrary many functions can be sampled with the generator, the following procedure was used to select functions that provide most information about the domain. The first function was randomly sampled and added to the sources F^S. Its local optima were identified by 121 local optimizations starting from different points θ^{start}. Afterward, 10 candidate functions from the problem domain were sampled. From this candidate set, the function with the lowest values for the best parameters that have been identified for each function in F^S was selected, analyzed and added to F^S. This selection is based on the assumption that a function for which good parameters of the current source functions F^S have a bad performance differs from the other sources and its information should be added to the model. The procedure was repeated until 10 source functions were selected and analyzed.

The solution space model was generated based on the identified local optima from all sources F^S and then used for the optimization on target functions f^T. Target functions were sampled from the same problem domain. CMA-ES [3] and Powell's Conjugate Gradient [4] were used as local optimizers. Their start points θ^{start} were sampled based on areas of the model. The model was also used to initialize the covariance setting of the CMA-ES. Both algorithms were compared to uninformed searches where the start points θ^{start} were randomly selected from the entire parameter space.

The procedure was tested on a two-dimensional (Fig. 1, a) and a six-dimensional problem domain. Models of both domains could be successfully identified (Fig. 1, c). The models improve the performance of optimizations on target functions f^T compared to uninformed optimizations (Fig. 2). This effect is stronger for the higher dimensional problem domain (Fig. 2, b). Further experiments showed that optimizations are also possible in cases where a non-fitting model is used, but the performance is reduced compared to an uninformed search.

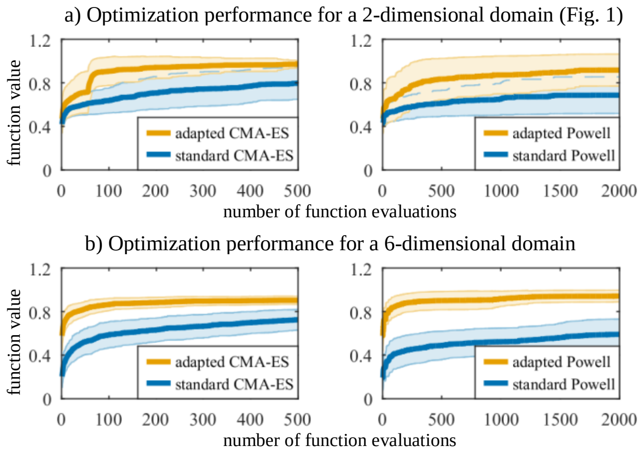

Fig. 2. Adapted versions of the CMA-ES and Powell's algorithm outperform standard versions in two different problem domains. Optimizations were performed on 100 unseen target functions of each problem domain. Plots show the mean value and the standard deviation over all targets for the best parameter that was found until the corresponding number of function evaluations.

IV. Conclusion

The proposed transfer learning method learns successfully a solution space model about the location of local optima for a problem domain from source tasks. The model then improves optimization of a target problem of the same domain. The approach is independent of the optimization algorithm that is used and can be used for any problem where source functions can be easily attained and evaluated. This makes the method applicable for a wide range of algorithms and problems.

References

[1] M. Ankerst, M.M. Breunig, H.P. Kriegel, and J. Sander. „Optics: Ordering points to identify the clustering structure", In Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data, volume 28, pages 49–60. ACM, 1999.

[2] M. Gallagher and B. Yuan. "A general-purpose tunable landscape generator", In IEEE Transactions on Evolutionary Computation, 10(5):590-603, 2006.

[3] N. Hansen. "The CMA Evolution Strategy: A Comparing Review", In Towards a new evolutionary computation: advances in the estimation of distribution algorithms, pages 75-102. Springer, 2006.

[4] M.J.D. Powell. "An effcient method for finding the minimum of a function of several variables without calculating derivatives", The Computer Journal, 7(2):155-162, 1964.