

# Visualizing fitnesses and constraint violations in single-objective optimization

Tomofumi Kitamura  
kitamura-tomofumi094@g.ecc.u-tokyo.ac.jp  
The University of Tokyo  
Tokyo, Japan

Alex Fukunaga  
fukunaga@idea.c.u-tokyo.ac.jp  
The University of Tokyo  
Tokyo, Japan

## ABSTRACT

In this study, we investigated visualization of search behavior in single-objective optimization function, where the objective function is composed of distinct components, either explicitly (as terms in the objective function or as components of a hybrid function) or implicitly (as constraints). We proposed a visualization method for constrained single-objective optimization in which the constraint violations and the term-by-term values of the polynomial objective function are separately calculated by RadViz and plotted in 3D. The proposed method is superior to the two-dimensional RadViz visualization in that it shows degasement of the fitness and constraint violations over time in the benchmark problem and can display them separately. Similarly, for the hybrid function, which is a benchmark problem consisting of multiple terms with different objective subfunctions, the difference in the timing of the decrease in fitness for each term is visualized by RadViz.

## CCS CONCEPTS

• **Human-centered computing** → **Visualization techniques**; • **Theory of computation** → *Evolutionary algorithms*; • **Computing methodologies** → *Continuous space search*.

## KEYWORDS

RadViz, Constrained single-objective optimization problem, Real-world optimization problem, Hybrid function

## ACM Reference Format:

Tomofumi Kitamura and Alex Fukunaga. 2021. Visualizing fitnesses and constraint violations in single-objective optimization. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion)*, July 10–14, 2021, Lille, France. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3449726.3463157>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*GECCO '21 Companion*, July 10–14, 2021, Lille, France

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-8351-6/21/07...\$15.00

<https://doi.org/10.1145/3449726.3463157>

## 1 INTRODUCTION

Understanding the behavior of optimization algorithms in real-world optimization problems with many constraints and complex evaluation functions is important but difficult, and various visualization methods have been proposed. For example, [16] uses heatmaps to visualize the exploration of real-world problems. In particular, there are several studies on visualization of solution sets in multi-objective optimization, as described in the next section. However, previous work on search visualization has not focused on the components of the objective function. We propose a method for visualizing constraint violations and fitnesses simultaneously, and present the results of visualizing the transition of the values of the fitness function components for each term during the search with DE and CMA-ES based algorithms on the CEC2020 Real-World Single Objective Constrained Optimization (CEC20RW) [13] Real-World benchmark problems. The proposed method is able to visualize the decrease of the fitness during the search of reducing the amount of constraint violation.

We also applied the visualization method to hybrid functions, which have similarities with the Real-World problem in that the fitness is based on the sum of several terms of different objective subfunctions. Hybrid function is a special setting in which there are no dependencies between the component functions, but the study of the behavior of search algorithms in this problem is very limited. We used RadViz to visualize the balance of the decrease in the three to six different component function values that make up the hybrid functions. This revealed that in the problems of CEC2017 Competition on Single Objective Bound Constrained Real-Parameter Numerical Optimization [2], some component functions such as Bent Cigar func tended to be solved first.

## 2 PREVIOUS WORK

The visualization of solutions in multi-objective optimization has been studied by projecting the Pareto front into two dimensions, such as [7, 21]. [21] shows a sorted heatmap and visualization. RadViz [9] is a multidimensional visualization method, where each dimension is equally spaced on the circumference of a circle, and points are plotted inside the circle. The closer a point is to the point that represents a particular dimension of the circumference, the larger the value of that dimension is compared to the other dimensions, and conversely, when the values of each dimension are almost equal, the point is plotted near the center of the circle. [1, 10, 15] extends RadViz as follows. 3D-RadViz [10] adds the distance

between the hyper-plane and the solution obtained from the extreme points to the usual 2D RadViz in the Pareto front visualization. Viz3D [1] tries to improve the viewability of RadViz by taking the average value of all variables on the z-axis. RadVizS [15] takes the distance from the origin of the original space on the z-axis and projects the data in three dimensions. In addition, Walker et al. [20, 22], visualized the search process to compare the characteristics of the problem, the differences between algorithms, and the effects of parameter settings. In the visualization of the population, visualization that connects the search space and the objective space is being worked on.[11, 21]

### 3 VIZUALIZATION OF SEARCH BEHAVIOR ON CEC20RWSO

The CEC2020 Real-World Single Objective Constrained Optimization (CEC20RWSO) benchmarks is a set of single-objective optimization benchmark problems which includes many kinds of real-world problems, including industrial chemical processes and process synthesis, design problems and mechanical engineering problems. In multi-objective optimization, individual fitnesses are recorded and Pareto fronts are often drawn, but in single-objective optimization, the fitnesses of each component of the objective function are rarely recorded and visualized. In this study, we analyze the fitness of each term explored by a single-objective optimization algorithm, and propose a method for visualizing the search process in constrained single-objective optimization.

#### 3.1 Experimental Setup

In the original CEC20RWSO benchmark code, each component of the objective function is added to a single fitness value, and only the summed value is returned. Algorithms minimize the summed value. To analyze the behavior of each component, we instrumented the code for functions F1 to F33 and expanded them into multiple terms to the extent possible, except for F4, F22, F24, F27, F28, and F31, whose expressions could not be divided into terms, and the values of all terms were recorded each time the best member in the population is updated. Details are given in Appendix. This instrumentation does not affect the behavior of the search algorithm on any of these functions.

COLSHADE[5] and sCMaGES[8], which came in second and third place, respectively, in the CEC20RWSO competition, were used in the experiments. COLSHADE is a variant of LSHADE[18] for constrained optimization, an algorithm that prioritizes the comparison of constraint violations over fitness in individual selection based on the constraints handling method [4]. sCMaGES is a CMA-ES based algorithm that performs gradient-based repair method [3]. The parameters are based on the original settings and the maximum number of evaluations is based on the competition rules. The source codes were downloaded from <https://github.com/P-N-Suganthan>.

#### 3.2 Standard Visualization Approach

The standard approach for visualization search algorithms is a performance plot such as Figure 1: for each term  $y_i$  in the additive fitness function  $y = y_0 + y_1, \dots$ , plot x-axis: number of evaluations, y-axis: normalized value of  $y_i$ , 1 line per term. First, we look at the results of COLSHADE. Since there are some problems in Fig. 1, such as F12, F15, F18 and F23, where the plots are crowded and difficult to read, we have also included both logarithmic plots in Fig. 2. F8 and F10 were decomposed into two terms, but they were excluded because one of their terms did not show any variation during the search. Each point represents an update of the best value in the population.

The results of sCMaGES are also shown in Figure 3. In Figure 4, only the y-axis is on a logarithmic scale. sCMaGES has frequent restarts, unlike COLSHADE, so instead of showing a point each time the best value of the population is updated, we show a point for the best member for each generation, resulting in a more crowded plot than the COLSHADE plots.

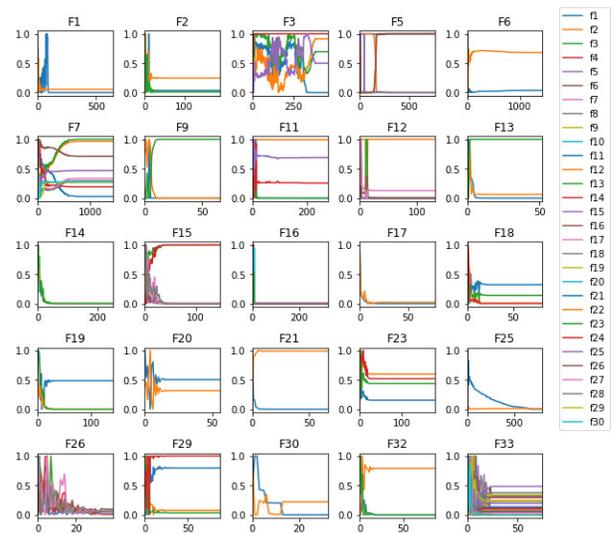


Figure 1: Visualization of the component function value trends obtained from the COLSHADE search using the standard method of showing a time series on the horizontal axis and a normalized fitness on the vertical axis.

#### 3.3 Using RadViz to visualize component function value trends

Figure 5 and Figure 6 are RadViz plots of same runs as in Figures 1-3. Each point on the circumference represents a component function value, and the color indicates the time series. The color is blue at the beginning of the search and becomes red as the search progresses. The RadViz coordinates were calculated using the Python library `Pandas.plotting.radviz`.

Looking at Figure 5, RadViz only has information about the relative size of each term, but it shows the changes that occurred during the search in a clear way. The plot of F3 in

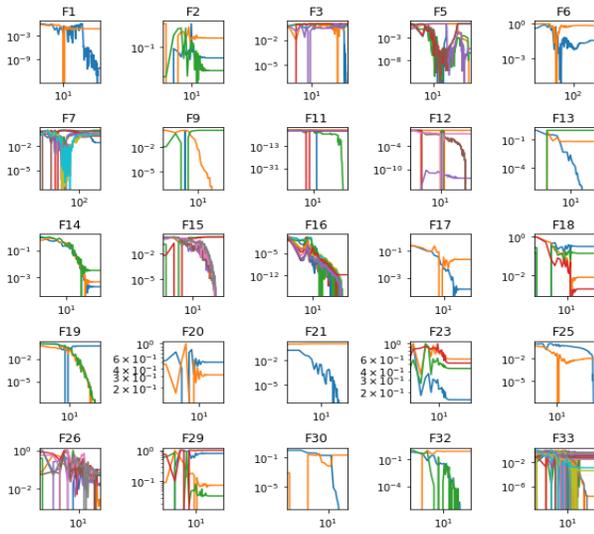


Figure 2: Figure 1 with both axes plotted on a logarithmic scale.

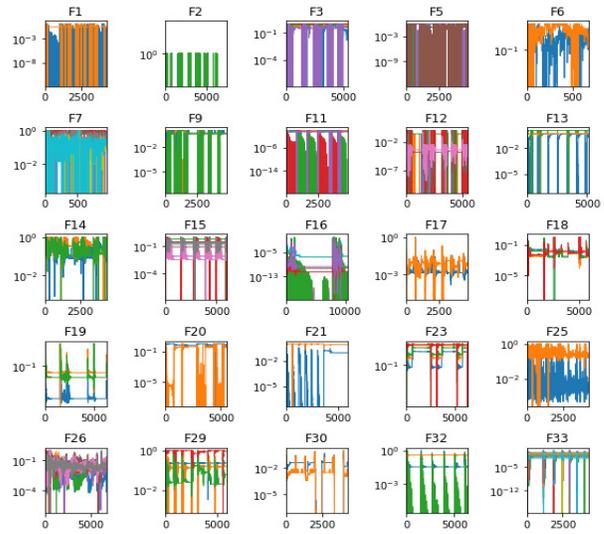


Figure 4: Same data as Figure 3, y-axis logarithmic scale.

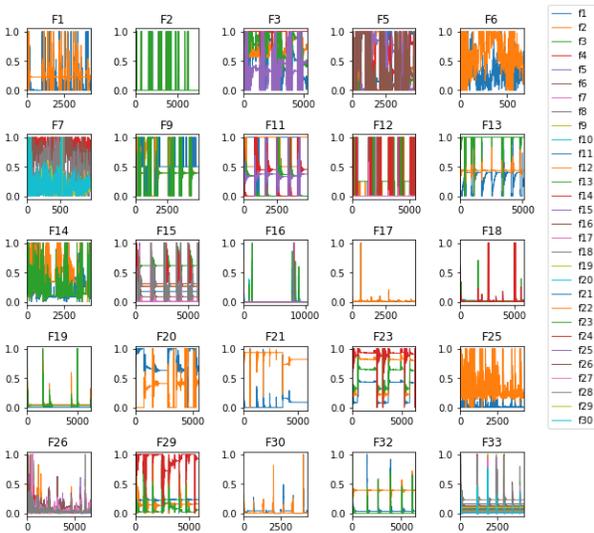


Figure 3: Visualization of the component function value trends obtained from the sCMaGES search using the standard method of showing a time series on the horizontal axis and a normalized fitness on the vertical axis.

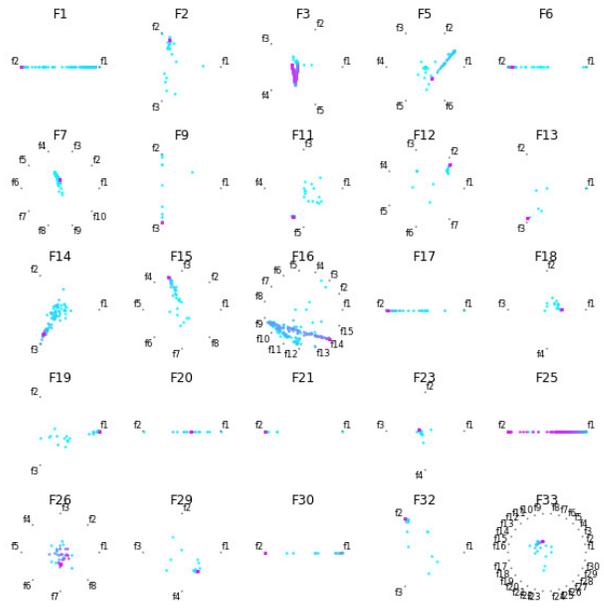


Figure 5: Visualization of the same data as in Figure 1 using RadViz.

the figure is a broken line, and we can see that it corresponds to the trade-off between  $f_3$  and  $f_5$  and the increase/decrease of  $f_2$  shown in Figure 1.  $F_5$  is hard to see in the performance plot because of the many colors, but upon close inspection, it can be seen that fitness decreases leaving  $f_2$  and  $f_5$ , then  $f_5$ , followed by  $f_2$ , and finally  $f_6$  increases.; RadViz simply shows the plot moving towards  $f_2$  and then back towards  $f_6$ .  $F_9$  in RadViz figure shows the plot moving from  $f_2$  to  $f_3$ , which is also easy to see from the performance plot. Similarly, the

plot ends at a particular term in  $F_{11}$ ,  $F_{12}$ ,  $F_{13}$ , and  $F_{32}$ , which can also be seen from the performance plot. In  $F_{15}$  and  $F_{29}$ , RadViz shows that the plot is directed to the middle of two terms, but in the performance plot of  $F_{15}$ , the lines overlap, and the colors of the lines are difficult to distinguish. The RadViz figure for  $F_{16}$  is the most distinctive, with the plot going from  $f_{12}$  to  $f_9$  and then reaching  $f_{14}$ . The 2 shows that the green and red lines are floating, but it is difficult to read the 15-color plot accurately. In general, although the information available from RadViz is limited, the variation

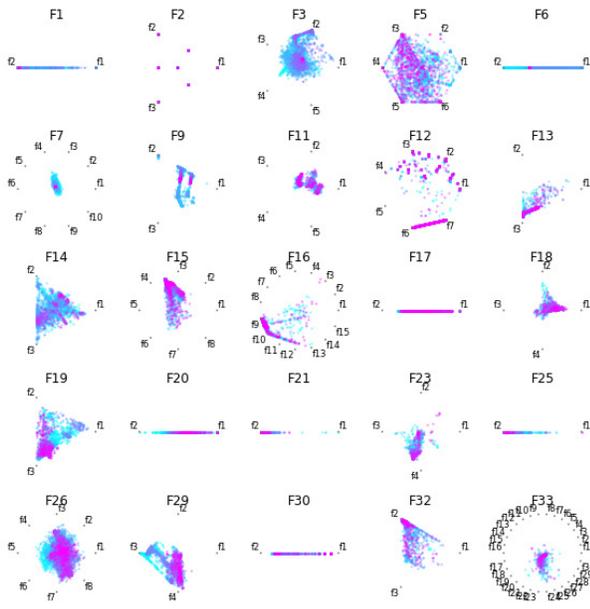


Figure 6: Visualization of the same data as in Figure 3 using RadViz.

in the timing of the changes in the fitness of each term is concisely expressed even if it is difficult to distinguish the colors of each term.

Next, we show the results of CMAgES. The RadViz plot in Figure 6 shows the time series as a color map, but the colors do not correspond to the number of generations because of the restart. Also, the plots are more widely distributed than COLSHADE, partly due to the large number of points plotted. From the figure, it can be seen that the restarts resulted in several clusters that are slightly different. For example, four lines are visible near the center of F9 in the figure. There are also a few lines near the center of F3 and F11, although they are slightly obscured. These are different from the lines linked to the search phase as seen in Figure 5. Figure 3 gives a more detailed fitness history, but the RadViz plot is clearer for F3, F9, and F29, where a few lines can be seen in the figure.

### 3.4 Visualization of constraint violation and component function value trends

To our knowledge, there has been no previous work in visualizing constraint violations during search by an evolutionary optimization algorithm. As a measure of constraint violation, the CEC20RWSO Competition uses the *feasibility rate*, which is the fraction of runs in which at least one feasible solution is attained. The reasons why it is difficult to simultaneously visualize the amount of constraint violation and the fitness is that the amount of constraint violation becomes zero when a feasible solution is obtained, and the number of constraint equations and the number of objective functions add up to an overcrowded circumference, which hinders interpretation.

A figure of constraint violations and objective functions arranged on a single circumference is shown in 7. The label *f* in the figure stands for fitness, *g* for inequality constraints, and *h* for equality constraints. The inclusion of the constraint violation amount, which approaches zero faster than fitness, makes Figure F7 more visible than Figure 5, which visualizes only fitness, because the points have shifted, but the fact that fitness and constraint violation are in the same plane makes it difficult to understand them separately. Also, since the amount of constraint violation has higher priority than fitness in COLSHADE, these should be separated to understand the behavior of the algorithm.

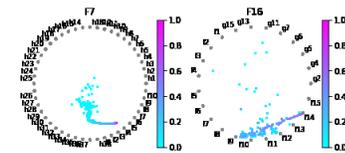


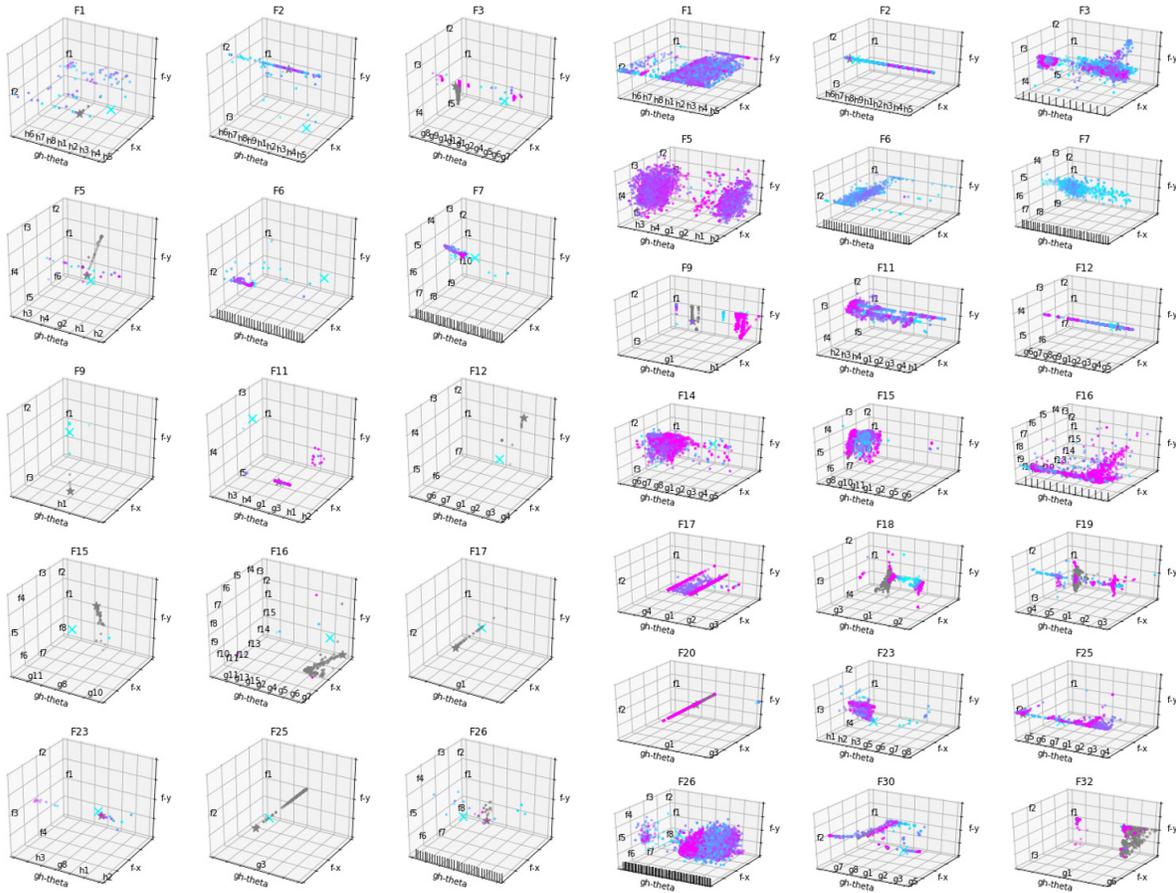
Figure 7: RadViz plot of constraint violations and component function values on the circumference. The color gradient indicates the time series.

### 3.5 Proposed method for simultaneous visualization of violations and objective values

We propose a method to visualize constraint violations and objective functions by distinguishing them on different axes, calculating coordinates by RadViz for each component function value and the amount constraint violation, and visualizing them in four dimensions: 3D space and color. The radius of the polar coordinates of the constraint violation expressed in RadViz is the third dimension. The radii of the polar coordinates correspond to the color map. The feasible solution with zero constraint violation is shown in gray, which is not in the color map. Constraints with zero constraint violation from the beginning of the search are not plotted. As with fitness, the amount of constraint violation was normalized to the range of 0 to 1 using the minimum and maximum values during the search.

### 3.6 Experimental Results

The results of COLSHADE are shown in the left column of Figure 8; the problems from F1 to F33 for which feasible solutions were obtained from the beginning of the search are not shown. The figure shows a compact visualization of how constraint violations are resolved, and component function values are optimized as the search proceeds. For F1, the plot is located in the plane because the objective function is decomposed into two terms. The plots are widely distributed, and we can see that some of them are in the red color, i.e., in an arrangement that is close to the circumference when the amount of constraint violation is expressed in RadViz. For F2, unlike F1, the plot is concentrated on *f2*, which indicates that *f1* and *f3* were reduced first, and the color of



**Figure 8: Visualization of constraint violations and component function values obtained by COLSHADE(left figure) and sCMaGES (right figure) using the proposed method. The coordinate axes represent the RadViz coordinates of the component function values and the radians of the RadViz representation of the amount of constraint violations. The colors correspond to the radii from 0 to 1, with red points having larger values and blue points having smaller values. Feasible solutions are shown in gray. The earliest and last points in the plot are marked with a cross and a star, respectively. Note that in some of the figures, the labels of constraint equations that were overcrowded were omitted due to paper limitations.**

the plot indicates that there was a stage where the first half of the nine constraint equations remained in violation of the constraints. From the plot of F6, we can read that f1 becomes smaller faster than f2 before the constraint violation becomes zero, and the remaining constraint violation is in the process of being solved in the next stage. For F7, we can see from the color of the plot that some constraint violations remain toward the end of the search. This is not readily apparent from Fig. 7, indicating the superiority of the proposed method. From some of the problems in the figure, we can also read the transition of fitness after a feasible solution is obtained. For F3, there was no bias in the decrease of fitness until some constraint equations were solved late, but after the feasible solution was obtained, we can see that the fitness decreased first except for f5. Similarly for F5, there is no significant change in the RadViz coordinates of fitness until the constraints are satisfied. On the other hand, in F16, after

the constraint was satisfied, the coordinates of fitness moved from f11 and f9 to f14. We can see that g7 was the last remaining constraint violation. Thus, the proposed method allows us to understand which constraint violations remain separately from the fitness, while showing the readable timing of the decrease in individual fitness and the decrease in the amount of constraint violations. However, there is a drawback: by using RadViz radians as the axes of the 3D plot, the constraint equations that used to be next to each other are now separated at both ends of the diagram. This is a point that requires attention when interpreting the figure, and we would like to improve this point by rearranging the constraint equations.

The results of sCMaGES are also shown in the right column of Figure 8. In F11, there was a shift of the plot from f1 to f5 in COLSHADE, and the plot is distributed from f1 to the center in sCMaGES as well. The clusters due to restarts

seen in RadViz can be seen more clearly in the figure with the added dimension of the constraint equation. In F18, we can see from this figure that three stages are connected: the stage where constraint g2 is violated and f4 remains, the stage where constraint g1 and f2 remain, and the stage where constraint g1 and f4 remain, which successfully visualizes the relationship between constraint violation resolution and fitness reduction. Similarly, in F30, the diagram suggests that there is a relationship between g8 and f1, and g5 and f2. In addition, overall, there are more plots that are spread out in the gh-theta axis in sCMaGES compared to COLSHADE. The continuous plots in the gh-theta direction seen in F12, F16, F17, F18, F19, and F30 represent the movement of plots such as g1 to g2 in F17 and g1 to g2 in F18 in the RadViz plot of the amount of constraint violation. In this chapter, we have clearly visualized that in CEC20-RW, some constraints or fitness terms are solved first in the search process. In the next section, we show that by decomposing and visualizing the terms of the fitness for another benchmark, each term of the fitness is solved in turn, as in CEC20RW.

## 4 BEHAVIOR OF SEARCH IN HYBRID FUNCTIONS

The hybrid function is an objective function that consists of multiple terms with different shapes similar to the Real-World problems we have seen so far. In this section, we tackle the visualization of the search process in which a particular component function value decreases first, which we also focused on in the previous section in the optimization of hybrid functions.

A hybrid function is a composite function created by dividing the problem dimension into multiple sets and assigning a different objective function to each set. Since each set is assigned variables in a specified ratio and a rotation matrix is applied to the group of variables, all the component functions have the property of being non-separable. However, since there is no dependency between the objective functions, the hybrid function is partially separable. Partial separability is a property that is also present in real problems [14]. [19], which studied the behavior of parameter adaptation when an adaptive DE searches for a two-component hybrid function, experiments showed that, depending on the proportion of variables assigned to each set, the adaptation of parameters to one component function at the expense of the other can lead to failure of search in some cases.

### 4.1 Visualization of fitness trends using RadViz

The experiments were conducted on four algorithms: EBOwith-CMAR [12], the first-place algorithm in the Competition on Single Objective Bound Constrained Real-Parameter Numerical Optimization [2], and the top three algorithms in the following year's competition using the same problem, HS-ES [23] (1st), LSHADE-RSP [17] (2nd), and ELSHADE-SPACMA [6] (3rd). The source codes were downloaded from <https://github.com/P-N-Suganthan>. The experimental setup

was 48 trials each with a maximum number of evaluations of dimensionality  $D \times 10000$ .

First, the fitnesses of component function at the end of the search in 30 dimensions is shown in the histogram 9. The histogram values for each component function were calculated by subtracting the average value of the best algorithm from the average value of each algorithm. The same color was used for identical component functions in the drawing. According to the figures, there is no difference among the algorithms for some of the component functions, suggesting that they are solved early in the search. On the other hand, Schwefel's func occupies a large proportion of the F12, F16, F17, and F20 figures, and is still in the process of minimization at the end of the search.

We also used RadViz to visualize how the component functions are solved in order. The value to be displayed is calculated as the degree to which the ratio of individual component function values to the total fitness deviates from  $1/N$  (where  $N$  is the number of components). This indicator is  $\frac{\text{each fitness}}{\text{sum of fitnesses}} - \frac{1}{N}$  and is the same as the formula for the mean error.

The results are shown in Figure 10. Each point is an output with a timing of  $(0.01, 0.02, 0.03, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0) \times \text{MaxEFeS}$  to be listed in the CEC18 Competition submission file. The figure shows that the plot moves to visit some functions on the circumference. There is a light blue plot near the Bent Cigar func and Elliptic func on the circumference, and the plot moves away from them. This indicates that these two functions will be solved first. In F11, F14, F15, F19, and F20, the plot visits the Rastrigin func approximately second. On the other hand, there is no plot near Ackley's func in F14, F17, and F20 of the figure. Only for F18, the plot ends between Ackley's func and Rastrigin's func, which means that it may vary depending on the combination of the constituent functions. Schwefel's func is visited last in F12, second in F16 and F17, and first in F20. One of the reasons why common functions are solved first in multiple problems is that the component functions are added together with fixed weight 1 for all terms. Therefore, for F14, F18, and F20, we tested multiplying one of the component functions by a weight. In the experiment, we multiplied the fitness of Schaffer's func by 1000 for F14, by 1/10 for Ackley's func for F18, and by 1/10 for Modified Schwefel's func for F20. The results after the modification are shown in Figure 11. Comparing the results before and after the modification, we can see that there is a change in the component functions whose weights were adjusted, as well as a relative change in the other component functions. Thus, we showed that the emphasis of the search changes depending on the weights applied to the components. We also found that same component functions are solved first across multiple problems in the CEC benchmark problem with weights fixed to 1.

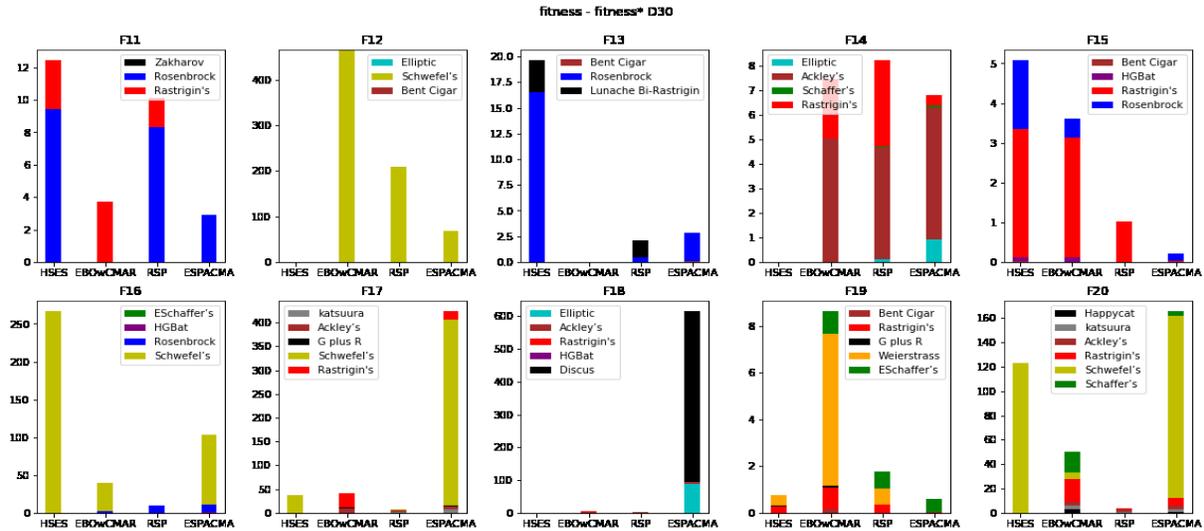


Figure 9: Stacked histograms of fitness by component function at the end of the search for the hybrid function in 30 dimensions. Bars are from left to right: HS-ES, EBOWwithCMAR, LSHADE-RSP, ELSHADE-SPACMA. The fitness per component function is the average value per algorithm minus the average value of the best algorithm.

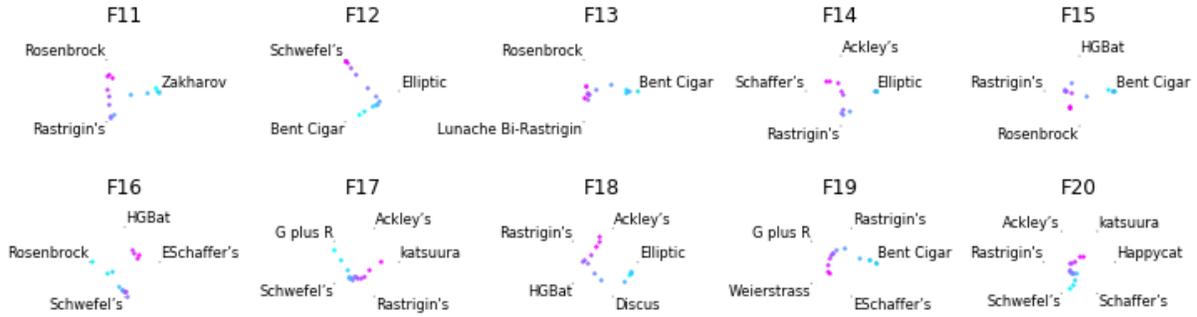


Figure 10: RadViz plot of component function values in 50 dimensions. The color gradient indicates the time series.



Figure 11: RadViz plots of F14, F18, and F20 after modification. The data before the modification of fitness is shown in gray for comparison.

## 5 CONCLUSION

In this study, we investigated visualization of search behavior in single-objective optimization function, where the objective function is composed of distinct components, either explicitly (as terms in the objective function or as components of a hybrid function) or implicitly (as constraints). We proposed a visualization method for constrained single-objective

optimization in which the constraint violations and the term-by-term values of the polynomial objective function are separately calculated by RadViz and plotted in 3D. The proposed method is superior to the two-dimensional RadViz visualization in that it shows the time points of decrease of the fitness and constraint violations in the benchmark problem and can display them separately. Similarly, for hybrid functions, which are a benchmark problem consisting of multiple terms with different objective subfunctions, the difference in the timing of the decrease in fitness for each term is visualized by RadViz. By doing so, we showed that there tends to be an ordering in which the component functions are optimized in the CEC2017 Competition hybrid function problems. In future research, we would like to improve the proposed method, such as considering an appropriate ordering of constraints, and apply it to multi-objective optimization.

REFERENCES

[1] Almir Olivette Artero and Maria Cristina Ferreira de Oliveira. 2004. Viz3d: Effective exploratory visualization of large multidimensional data sets. In *Proceedings. 17th Brazilian Symposium on Computer Graphics and Image Processing*. IEEE, 340–347.

[2] N.H. Awad, M.Z. Ali, J.J. Liang, B.Y. Qu, and P.N. Suganthan. 2016. *Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Bound Constrained Real-Parameter Numerical Optimization*. Technical Report. Nanyang Technological University, Singapore.

[3] Piya Chootinan and Anthony Chen. 2006. Constraint handling in genetic algorithms using a gradient-based repair method. *Computers & operations research* 33, 8 (2006), 2263–2281.

[4] Kalyanmoy Deb. 2000. An efficient constraint handling method for genetic algorithms. *Computer methods in applied mechanics and engineering* 186, 2-4 (2000), 311–338.

[5] Javier Gurrola-Ramos, Arturo Hernández-Aguirre, and Oscar Dalmau-Cedeño. 2020. COLSHADE for Real-World Single-Objective Constrained optimization Problems. In *2020 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 1–8.

[6] Anas A Hadi, Ali W Mohamed, and Kamal M Jambi. 2021. Single-objective real-parameter optimization: Enhanced LSHADE-SPACMA algorithm. In *Heuristics for Optimization and Learning*. Springer, 103–121.

[7] Zhenan He and Gary G Yen. 2015. Visualization and performance metric in many-objective optimization. *IEEE Transactions on Evolutionary Computation* 20, 3 (2015), 386–402.

[8] Michael Hellwig and Hans-Georg Beyer. 2020. A modified matrix adaptation evolution strategy with restarts for constrained real-world problems. In *2020 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 1–8.

[9] Patrick Hoffman, Georges Grinstein, Kenneth Marx, Ivo Grosse, and Eugene Stanley. 1997. DNA visual and analytic data mining. In *Proceedings. Visualization'97 (Cat. No. 97CB36155)*. IEEE, 437–441.

[10] Amin Ibrahim, Shahryar Rahnamayan, Miguel Vargas Martin, and Kalyanmoy Deb. 2016. 3D-RadVis: Visualization of Pareto front in many-objective optimization. In *2016 IEEE Congress on Evolutionary Computation (CEC)*. 736–745. <https://doi.org/10.1109/CEC.2016.7743865>

[11] Maki Kubota, Takayuki Itoh, Shigeru Obayashi, and Yuriko Takeshima. 2014. EVOLVE: A visualization tool for multi-objective optimization featuring linked view of explanatory variables and objective functions. In *2014 18th International Conference on Information Visualisation*. IEEE, 351–356.

[12] Abhishek Kumar, Rakesh Kumar Misra, and Devender Singh. 2017. Improving the local search capability of effective butterfly optimizer using covariance matrix adapted retreat phase. In *2017 IEEE congress on evolutionary computation (CEC)*. IEEE, 1835–1842.

[13] Abhishek Kumar, Guohua Wu, Mostafa Z Ali, Rammohan Mallipeddi, Ponnuthurai Nagaratnam Suganthan, and Swagatam Das. 2020. A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. *Swarm and Evolutionary Computation* 56 (2020), 100693.

[14] Zhi-Pei Liang. 2007. Spatiotemporal imaging with partially separable functions. In *2007 4th IEEE international symposium on biomedical imaging: from nano to macro*. IEEE, 988–991.

[15] Lenka Nováková and Olga Štepanková. 2009. Radviz and identification of clusters in multidimensional data. In *2009 13th International Conference Information Visualisation*. IEEE, 104–109.

[16] Andy Pryke, Sanaz Mostaghim, and Alireza Nazemi. 2007. Heatmap visualization of population based multi objective algorithms. In *International conference on evolutionary multi-criterion optimization*. Springer, 361–375.

[17] Vladimir Stanovov, Shakhnaz Akhmedova, and Eugene Semenkin. 2018. LSHADE algorithm with rank-based selective pressure strategy for solving CEC 2017 benchmark problems. In *2018 IEEE congress on evolutionary computation (CEC)*. IEEE, 1–8.

[18] Ryoji Tanabe and Alex S Fukunaga. 2014. Improving the search performance of SHADE using linear population size reduction. In *2014 IEEE congress on evolutionary computation (CEC)*. IEEE, 1658–1665.

[19] Ryoji Tanabe and Alex S Fukunaga. 2014. On the pathological behavior of adaptive differential evolution on hybrid objective functions. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*. 1335–1342.

[20] David J Walker and Matthew J Craven. 2018. Toward the online visualisation of algorithm performance for parameter selection. In *International Conference on the Applications of Evolutionary Computation*. Springer, 547–560.

[21] David J Walker, Richard M Everson, and Jonathan E Fieldsend. 2012. Visualizing mutually nondominating solution sets in many-objective optimization. *IEEE Transactions on Evolutionary Computation* 17, 2 (2012), 165–184.

[22] Mathew J Walter, David J Walker, and Matthew J Craven. 2020. Visualising Evolution History in Multi-and Many-Objective Optimisation. In *International Conference on Parallel Problem Solving from Nature*. Springer, 299–312.

[23] Geng Zhang and Yuhui Shi. 2018. Hybrid sampling evolution strategy for solving single objective bound constrained problems. In *2018 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 1–7.

A MODIFIED CEC20RW PROBLEMS

- F1  $f_1 = 35 \times x_1^{0.6}, f_2 = 35 \times x_2^{0.6}$
- F2  $f_1 = (\frac{x_1}{120 \times x_4})^{0.6}, f_2 = (\frac{x_2}{80 \times x_5})^{0.6}, f_3 = (\frac{x_3}{40 \times x_6})^{0.6}$
- F3  $f_1 = -1.715 \times x_1, f_2 = -0.035 \times x_1 \times x_6,$   
 $f_3 = -4.0565 \times x_3, f_4 = -10 \times x_2, f_5 = 0.063 \times x_3 \times x_5,$
- F5  $f_1 = -9 \times x_1, f_2 = -15 \times x_2, f_3 = 6 \times x_3,$   
 $f_4 = 16 \times x_4, f_5 = 10 \times x_5, f_6 = 10 \times x_6$
- F6  $f_1 = 0.9979 + 0.00432 \times x_5, f_2 = 0.01517 \times x_{13}$
- F7  $f_1 = c_{1,1} + c_{2,1} \times x_5, f_2 = c_{3,1} \times x_{24} \times x_5,$   
 $f_3 = c_{4,1} \times x_{28} \times x_5, f_4 = c_{5,1} \times x_{33} \times x_5,$   
 $f_5 = c_{6,1} \times x_{34} \times x_5, f_6 = c_{1,2} + c_{2,2} \times x_{13},$   
 $f_7 = c_{3,2} \times x_{26} \times x_{13}, f_8 = c_{4,2} \times x_{31} \times x_{13},$   
 $f_9 = c_{5,2} \times x_{38} \times x_{13}, f_{10} = c_{6,2} \times x_{39} \times x_{13}$
- F9  $f_1 = -x_3, f_2 = 2 \times x_1, f_3 = x_2$
- F11  $f_1 = 7.5 \times \text{round}(x_5), f_2 = 5.5 \times \text{round}(x_6),$   
 $f_3 = 7 \times x_3, f_4 = 6 \times x_4, f_5 = 5 \times x_7$
- F12  $f_1 = (\text{round}(x_4) - 1)^2, f_2 = (\text{round}(x_5) - 1)^2,$   
 $f_3 = (\text{round}(x_6) - 1)^2, f_4 = -\log(\text{round}(x_7) + 1),$   
 $f_5 = (x_1 - 1)^{22}, f_6 = (x_2 - 2)^2, f_7 = (x_3 - 3)^2$
- F13  $f_1 = -5.357854 \times x_1^2, f_2 = -0.835689 \times \text{round}(x_4) \times x_3,$   
 $f_3 = -37.29329 \times \text{round}(x_5) + 40792.141$
- F14  $f_1 = \text{alp} \times \text{round}(x_1) \times \text{round}(x_4)^{\text{beta}},$   
 $f_2 = \text{alp} \times \text{round}(x_2) \times \text{round}(x_5)^{\text{beta}},$   
 $f_3 = \text{alp} \times \text{round}(x_3) \times \text{round}(x_6)^{\text{beta}}$
- F15  $f_1 = 0.7854 \times x_1 \times x_2^2 \times (3.3333 \times x_3^2),$   
 $f_2 = 0.7854 \times x_1 \times x_2^2 \times (14.9334 \times x_3 - 43.0934),$   
 $f_3 = -1.508 \times x_1 \times (x_4^2), f_4 = -1.508 \times x_1 \times (x_7^2),$   
 $f_5 = 7.477 \times (x_6^3), f_6 = 7.477 \times (x_7^3),$   
 $f_7 = 0.7854 \times (x_4 \times x_6^2), f_8 = 0.7854 \times (x_5 \times x_7^2)$
- F16  $f_1 = 63098.88 \times x_2 \times x_4 \times x_{12}, f_2 = 5441.5 \times x_2^2 \times x_{12},$   
 $f_3 = 115055.5 \times x_2^{1.664} \times x_6, f_4 = 6172.27 \times x_2^2 \times x_6,$   
 $f_5 = 63098.88 \times x_1 \times x_3 \times x_{11}, f_6 = 5441.5 \times x_1^2 \times x_{11},$   
 $f_7 = 115055.5 \times x_1^{1.664} \times x_5, f_8 = 6172.27 \times x_1^2 \times x_5,$   
 $f_9 = 140.53 \times x_1 \times x_{11}, f_{10} = 281.29 \times x_3 \times x_{11},$   
 $f_{11} = 70.26 \times x_2^2, f_{12} = 281.29 \times x_1 \times x_3, f_{13} = 281.29 \times x_3^2,$   
 $f_{14} = 14437 \times x_8^{1.8812} \times x_{12}^{0.3424} \times x_{10} \times x_{14}^{-1} \times x_7 \times x_9^{-1},$   
 $f_{15} = 20470.2 \times x_2^{2.893} \times x_9^{0.316} \times x_1^2$
- F17  $f_1 = x_1^2 \times x_2 \times x_3, f_2 = x_1^2 \times x_2 \times 2$
- F18  $f_1 = 0.6224 \times x_1 \times x_3 \times x_4, f_2 = 1.7781 \times x_2 \times x_3^2,$   
 $f_3 = 3.1661 \times x_1^2 \times x_4, f_4 = 19.84 \times x_1^2 \times x_3$
- F19  $f_1 = 1.10471 \times x_1^2 \times x_2, f_2 = 0.04811 \times x_3 \times x_4 \times 14,$   
 $f_3 = 0.04811 \times x_3 \times x_4 \times x_2$
- F20  $f_1 = (2 \times \text{sqrt}(2) \times x_1) \times 100, f_2 = (x_2) \times 100$
- F21  $f_1 = \text{pi} \times (x_2^2) \times x_3 \times (x_5 + 1) \times \text{rho},$   
 $f_2 = \text{pi} \times (-x_1) \times x_3 \times (x_5 + 1) \times \text{rho}$
- F23  $f_i = \text{rho} \times (x_5 \times 1e^{-3}) \times \text{pi}/4 \times (x_i \times 1e^{-3})^2 \times (1 + (N_i/N)^2)$   
 $(i = 1, \dots, 4)$
- F25  $f_1 = (Q \times \text{Po}/0.7)/12, f_2 = \text{Ef}/12$
- F26  $f_i = \text{pi}/1000 \times b_j \times c_j^2 \times (Np_j^2)/(Np_j + Ng_j)^2$   
 $(i = 1, \dots, 8, j = i//2)$
- F29  $f_1 = 8.61 \times 1e5 \times x_1^{0.5} \times x_2 \times x_3^{-2/3} \times x_4^{-1/2},$   
 $f_2 = 3.69 \times 1e4 \times x_3,$   
 $f_3 = 7.72 \times 1e8 \times x_1^{-1} \times x_2(0.219), f_4 = -765.43 \times 1e6 \times x_1^{-1}$
- F30  $f_1 = (\text{pi}^2 \times x_2 \times \text{round}(x_3)^2 \times 2)/4$   
 $f_2 = (\text{pi}^2 \times x_2 \times \text{round}(x_3)^2 \times \text{round}(x_1))/4$
- F32  $f_1 = 5.3578547 \times x_3^2, f_2 = 0.8356891 \times x_1 \times x_5,$   
 $f_3 = 37.293239 \times x_1 - 40792.141$
- F33  $f(i + ni \times (j - 1)) = X(j, i)^{\text{pval}} \times Ue^i \times KE \times Ue$