A flexible dissimilarity measure for active and passive 3D structures and its application in the fitness–distance analysis [*]

Maciej Komosinski[0000−0001−8438−5574] and
Agnieszka Mensfelt[0000−0002−2385−2017]

Poznan University of Technology, Piotrowo 2, 60-965 Poznan, Poland
maciej.komosinski@cs.put.poznan.pl

Abstract. Evolutionary design of 3D structures – either static structures, or equipped with some sort of a control system – is one of the hardest optimization tasks. One of the reasons are rugged fitness landscapes resulting from complex and non-obvious genetic representations of such structures and their genetic operators. This paper investigates global convexity of fitness landscapes in optimization tasks of maximizing velocity and height of both active and passive structures. For this purpose, a new dissimilarity measure for 3D active and passive structures represented as undirected graphs is introduced. The proposed measure is general and flexible – any vertex properties can be easily incorporated as dissimilarity components. The new measure was compared against the previously introduced measure in terms of triangle inequality satisfiability, changes in raw measure values and the computational cost. The comparison revealed improvements for triangle inequality and raw values at the expense of increased computational complexity. The investigation of global convexity of the fitness landscape, involving the fitness–distance correlation analysis, revealed negative correlation between the dissimilarity of the structures and their fitness for most of the investigated cases.

Keywords: evolutionary design · 3D structure · dissimilarity measure · optimization · global convexity

1 Introduction

Three-dimensional structures can be divided into two main groups based on whether they have a system that actively controls their physical body, or they don't have such a system. Passive designs do not perform any action (they are static) and therefore only their physical structure exists and can be subject to

[*] This paper was published in *Applications of Evolutionary Computation*, LNCS vol. 11454, Springer, 2019, https://doi.org/10.1007/978-3-030-16692-2_8.
The second author was supported by the Faculty of Computing, Poznan University of Technology, through the funds provided by the Ministry of Science and Higher Education.

evolutionary optimization – examples of such structures are an antenna [1], a bridge [2,3] or a truss [4]. Conversely, active designs do have a control system, and the system is evolved along with the body – such structures are often encountered in robotics [5,6,7] and in artificial life [8,9].

Designing 3D structures can be considered one of the hardest optimization problems for a number of reasons: the search space is usually infinitely big, the optimization process must combine both continuous and discrete aspects, and evaluation of solutions is costly, non-deterministic, and involves multiple criteria [10]. Additionally, a complex, non-obvious representation of solutions along with its genetic operators usually leads to highly rugged fitness landscapes [11,12]. There are many metaheuristic approaches that can be used for automated design, but evolutionary algorithms turned out to be the most successful [11]. Despite the progress in the area of evolutionary design, many aspects have to be still improved before it will be possible to routinely apply this technique for successful automated design of 3D structures.

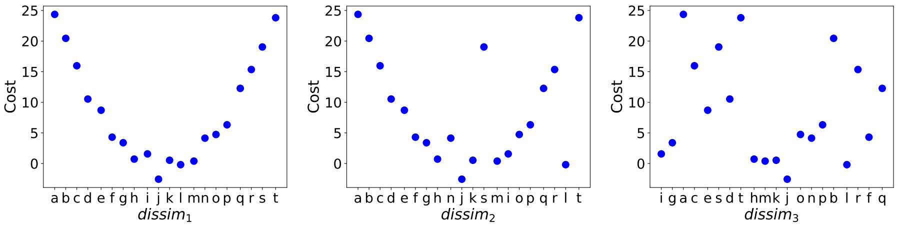


Fig. 1: Three sample fitness landscapes with the set of the same 20 solutions denoted by letters ‘a’ to ‘t’ (in this work, these letters correspond to different 3D structures). Each of these fitness landscapes is induced by a different dissimilarity measure, from $dissim_1$ – the most convex, to $dissim_3$ – the most chaotic. For best performance of the optimization algorithm and the exploitation of the global convexity property during search, the neighborhood and reconfiguration operators should preserve the topology induced by $dissim_1$.

The difficulty of any optimization task relies heavily on the shape of its fitness landscape, with rugged landscapes generally corresponding to substantially harder optimization tasks than smooth landscapes [13,14]. For combinatorial optimization problems (e.g., for the traveling salesman problem), it was demonstrated that their fitness landscape may possess the property of global convexity [15]. If the fitness landscape exhibits this property, better solutions are more similar to each other and to the global optimum than they are similar to worse solutions. This property can be exploited to facilitate the process of evolutionary search. Note that the fitness landscape may possess the property of global convexity or not – this depends on how the landscape is constructed, i.e., on genetic operators that are used to traverse the landscape [16] (see Fig. 1). In order to construct a smooth, globally convex landscape, efficient genetic operators need to be designed. This is where the dissimilarity measure is helpful: it allows to evaluate the correlation between the similarity of solutions and the similarity of their fitness values. If this correlation is high (i.e., the dissimilarity measure captured some important characteristics of solutions), then designing genetic operators such that neighboring solutions in the search space are similar will result in a smooth fitness landscape.

Despite the successes in the combinatorial optimization domain, global convexity was not extensively researched so far in the area of optimization of active and passive 3D structures. One of the reasons is the difficulty of dissimilarity calculation for such structures. Therefore, the first aim of this paper is the introduction of a new measure for active and passive 3D structures. The second aim is the application of this measure in the analysis of global convexity of fitness landscape for different optimization tasks, such as maximizing the velocity of active 3D structures, and maximizing the height of passive 3D structures. The results of such analyses may help design genetic representations and operators that will increase the efficiency of solving the demanding tasks of evolutionary design.

1.1 Related work

Global convexity can be assessed using the fitness–distance correlation (FDC), i.e., the correlation between the fitness value and the distance (i.e., dissimilarity) between solutions – for example the distance to the global optimum, if it is known. Fitness–distance correlation analysis was introduced by Jones and Forrest as a method of investigation of the difficulty of optimization problems [17]. As previously mentioned, global convexity depends on the specific distance (dissimilarity) measure used. Therefore, the FDC analysis using different dissimilarity measures can be employed to identify properties of solutions that correlate with their fitness, and then to devise genetic operators that preserve such properties. One example of such an operator is the distance-preserving crossover (DPX). Such an operator attempts to guarantee that the distance between each parent solution and a child solution is not higher than the distance between parents [18,19]. So far, global convexity tests were successfully used for the development of distance-preserving crossover operators in combinatorial optimization problems [20,21,22].

Apart from the FDC analysis, dissimilarity measures for 3D structures have plenty of other applications [10], including automated classification, discovering clusters in solutions, population analysis and inferring dendrograms. Dissimilarity measures for 3D structures already exist in many domains such as computer vision [23], bioinformatics [24] or chemical informatics [25], but they are usually too domain-specific to be applied to comparing arbitrary 3D designs. Additionally, the measure should take into account not only the "body" of the solution, but in the case of active structures, also their control systems. Moreover, the algorithm for calculating the measure needs to be efficient to handle complex 3D structures. The dissimilarity measure introduced by Komosinski and Kubiak [26] was designed to fulfill these demands, however it has a number of disadvantages described in Sect. 2.3.

2 Dissimilarity measure for active and passive 3D structures

The dissimilarity measure introduced in this paper was designed to overcome the problems of the simpler measure devised earlier [26]. The calculation algorithms of both measures, as well as a model of a 3D structure required by the measures, are described in the following sections. The proposed measure has been implemented in the Framsticks simulation environment [27,9] and the C++ source code is available as a part of the Framsticks SDK [28]. The value calculated by the algorithm for two structures is interpreted as follows:

- 0 means that both structures are identical (i.e., no dissimilarity),
- a positive value reflects the dissimilarity (the "distance") between both structures.

2.1 3D active structure model

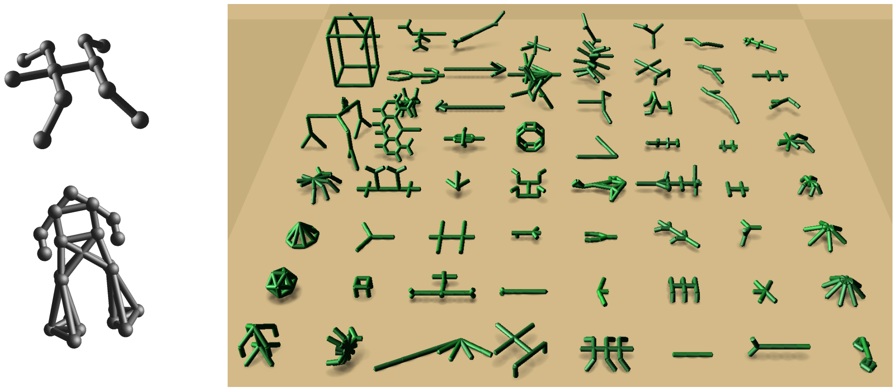


Fig. 2: Sample 3D structures compatible with the model considered in this work. Left: a close-up of two structures with visible vertices and edges. Right: an 8×8 sample of structures from the *w+s+o* test set described in Sect. 2.5.

The 3D structures considered in this work are modeled as undirected graphs (Fig. 2), and such structures are simulated in the Framsticks environment [27,9]. The structures can be either active or passive. Active structures are equipped with working control units – artificial neurons – including sensors and effectors. The neurons may be optionally attached to vertices. Therefore, each vertex in a structure can be described with the following properties:

- its degree (i.e., the number of edges incident to the vertex),
- the number of neurons attached to the vertex,
- its position in the three-dimensional coordinate system.

2.2 Dissimilarity measure assumptions

The measure should allow for the comparison of different properties of any two 3D structures. Therefore, the value of the measure consists of four components:

- d_V – the absolute difference in the number of vertices in both structures,
- d_D – the absolute difference in the degree of matched vertices,
- d_N – the absolute difference in the number of neurons attached to the matched vertices,
- d_G – the Euclidean distance between matched vertices.

Since in the simplest case these components are aggregated into a single value using the weighted sum, the user can adjust the importance of each component by setting the weight (w_V, w_D, w_N and w_G) of this component to a value higher than or equal to zero.

If the w_G weight is higher than zero, the two structures should be aligned in 3D space before calculating Euclidean distances between matched vertices. For this purpose, a multidimensional scaling [29] procedure (MDS) has been used for each structure separately. After the application of this procedure, centers of both structures are located in the origin of the coordinate system. The axis with the highest variance of coordinates is chosen as the first axis of the structure, and the axis with the second highest variance of coordinates is chosen as the second axis of the structure.

Since a standard MDS procedure takes as an input the distance matrix based on original vertex coordinates, vertex degrees are not considered during spatial alignment of structures. As a result, vertices with a similar vertex degree in both structures may not be aligned properly. To overcome this problem, instead of the standard MDS procedure, we use the weighted MDS (wMDS). In wMDS, during the alignment, the distance matrix between vertex coordinates is weighted using vertex degrees as weights. In this way, the information about vertex degrees is incorporated and exploited during the alignment process – this process uses richer information which ultimately leads to better alignment of the structures.

2.3 The original dissimilarity measure: vertex degree order and greedy matching

The dissimilarity measure proposed earlier [26] is a heuristic. Its algorithm can be divided into three main parts:

- Alignment of the structures,
- Construction of the matching function,
- Dissimilarity calculation.

The alignment procedure has been described in Sect. 2.2. The main part of the algorithm is the construction of the matching function. In order to build the matching, the vertices of each structure are sorted according to the vertex degree, and then according to the number of neurons within the groups of the same vertex degree. The matching procedure starts with vertices with the highest vertex degree in both structures, and tries to find the pairs of matching vertices. The within-group matching ends when there are no unmatched vertices with a given degree in one or both of the structures. Then the algorithm proceeds to handle the group of vertices with the second highest vertex degree, and then continues and handles groups with lower and lower vertex degree. Within groups with the same vertex degree, the matching is built according to the minimum distance calculated as

$$w_D \cdot d_D(v_{i1}, v_{j2}) + w_N \cdot d_N(v_{i1}, v_{j2}) + w_G \cdot d_G(v_{i1}, v_{i2})$$

where weight w_x corresponds to the x component of the measure, v_{i1} denotes the i-th vertex of the first structure, and v_{j2} denotes the j-th vertex of the second structure. For details of the procedure, see [26].

One problem of this measure is the fixed order in which the vertices are matched. Even if the weight of the d_D component is equal to zero, the matching procedure still starts from the vertices with the highest vertex degree and follows the logic described above. Another issue with the matching procedure is that it is a greedy algorithm. It always chooses the matching which provides the minimal distance between currently considered vertices, however this choice does not have to result in the minimal overall distance between complete structures. This algorithm will be referred to in the following sections as $dissim_{DegGreedy}$ (vertex degree order and greedy matching).

2.4 The improved dissimilarity measure: flexible criteria order and optimal matching

The dissimilarity measure proposed in this paper overcomes the disadvantages of $dissim_{DegGreedy}$ described above. The calculation of the improved dissimilarity measure is also preceded by the alignment procedure described in Sect. 2.2. In order to avoid the greediness of the matching procedure, the Kuhn-Munkres algorithm [30,31] (also known as the Hungarian algorithm) is applied. The goal of the matching procedure is to find the matching of vertices that will minimize the overall distance between two structures. The overall distance consists of two components: the sum of the distances between matched vertices, and the penalty for the unmatched vertices (only present in the case of the structures differing in the number of vertices). The distance between each pair of matched vertices is calculated as follows:

$$dist_{v_{i1}v_{j2}} = w_D \cdot d_D(v_{i1}, v_{j2}) + w_N \cdot d_N(v_{i1}, v_{j2}) + w_G \cdot d_G(v_{i1}, v_{i2})$$

The penalty for each unmatched vertex v_i is the sum of the following components:

- $penalty_D(v_i) = w_D \cdot \text{vertex\_degree}(v_i)$,
- $penalty_N(v_i) = w_N \cdot \text{number\_of\_neurons}(v_i)$,
- $penalty_G(v_i) = w_G \cdot \text{distance\_to\_the\_origin}(v_i)$.

In order to take the penalty into account during the Hungarian matching procedure, additional rows or columns are created in the distance matrix for the smaller structure. Such additional rows or columns are filled with the penalty for the inability to match the parts of the structure with more vertices. This dissimilarity measure will be referred to as $dissim_{FlexOpt}$ (flexible criteria order and optimal matching). The matching procedure along with the distance calculation procedure is outlined in Listing 1.1.

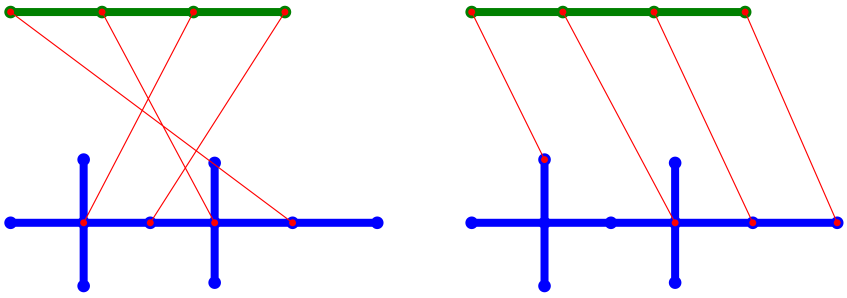
```
GS = structure with more vertices ("greater")
SS = structure with less vertices ("smaller")
nGreat = vertices_number(GS)
nSmall = vertices_number(SS)
dist_matrix = matrix(size=(nGreat, nGreat))
for i in range(0, nGreat-1):
  for j in range(0, nGreat-1):
    if i >= nSmall:
          dist_matrix[i][j] = penalty($v_{GSj}$)
        else:
          dist_matrix[i][j] = dist($v_{SSi}$,$v_{GSj}$)
matched = HungarianAlgorithm(dist_matrix)
distance = $\sum_i \sum_j$ dist(matched($v_{SSi}$,$v_{GSj}$))
distance = distance + difference_in_unattached_neurons(GS, SS)
distance = distance + $w_V \cdot d_V(GS, SS)$
```

Listing 1.1: The outline of the $dissim_{FlexOpt}$ vertex matching and dissimilarity measure calculation algorithm.

The measure described above is very general and can be used for calculation of dissimilarity of any 3D objects that can be represented as undirected graphs. Furthermore, $dissim_{FlexOpt}$ is more flexible than $dissim_{DegGreedy}$. In $dissim_{DegGreedy}$, some of the components were more important than others independently of the weight values – vertex degree and neuron count were always used to sort the vertices before matching. In $dissim_{FlexOpt}$, all of the components are processed in a uniform way. This allows any properties of the vertices in the model to be easily incorporated into dissimilarity calculation as subsequent measure components. The proposed measure will be referred to in the following sections as $dissim_{FlexOpt}$ (flexible criteria order and optimal matching). The matchings of the same two 3D structures obtained using $dissim_{DegGreedy}$ and $dissim_{FlexOpt}$ algorithms are compared in Fig. 3.

2.5 Comparison of dissimilarity measures

Both measures, the original one ($dissim_{DegGreedy}$) and the one introduced in this paper ($dissim_{FlexOpt}$), were compared on four different sets of three-dimensional



(a) $dissim_{DegGreedy} = 51.55$ (b) $dissim_{FlexOpt} = 42.58$

Fig. 3: The matching and dissimilarity between the same two structures obtained using $dissim_{DegGreedy}$ and $dissim_{FlexOpt}$ measures with all the weights equal, $w_V = w_D = w_N = w_G = 1$. The $dissim_{DegGreedy}$ matching procedure (a) starts from vertices with the highest vertex degree in both structures. Therefore, vertices with the degree of 4 from the bottom structure have to be matched with vertices with the degree of 2 from the top structure, and then vertices with the degree of 2 from the bottom structure have to be matched with vertices with the degree of 1 from the top structure. In the case of $dissim_{FlexOpt}$ (b), the matching procedure tries to find the matching that minimizes the total dissimilarity value.

models using three evaluation criteria. For each of the test sets, the dissimilarity was calculated for all pairs of the structures in the set – for example, for a set with 400 structures, the dissimilarity was computed $400 \times 400 = 160\,000$ times.

Test sets. Four test sets of 3D structures were used. Three of these sets (*walking*, *swimming* and *other*) are sample sets provided in the Framsticks distribution [27] and consist mostly of structures designed by humans, or pre-designed manually and then evolved to meet some specific goal. The fourth set (*best400*) is the result of an evolutionary experiment, with four fitness criteria and 100 structures optimized for each criterion using evolutionary algorithms [10] (the visualization of this experiment is available at https://www.youtube.com/watch?v=lo4vL7gOuYk). The test sets differ in the number of structures they contain and in the size (i.e., the number of vertices) of the structures. The distribution of the structure size in each test set is shown in Fig. 4.

Comparison criteria and results. Both measures – $dissim_{DegGreedy}$ and $dissim_{FlexOpt}$ – were compared using three criteria; these criteria are described below.

Triangle inequality violation. It is desirable for a dissimilarity measure to be a metric – with this property, it is possible to construct a metric space for the set

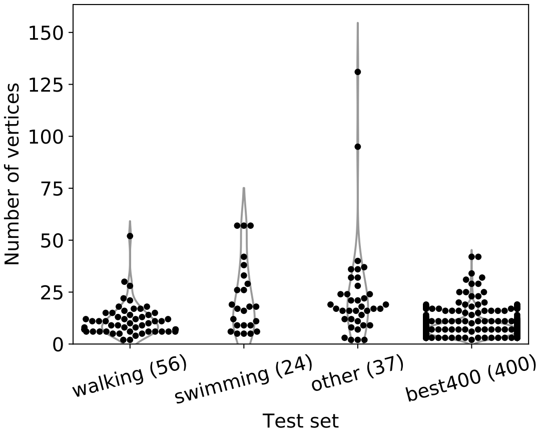


Fig. 4: The distribution of the number of vertices in 3D structures for each test set. The number in parentheses after the name of the test set indicates the number of structures in that set.

of analyzed structures. In order to be a metric, the measure must satisfy the conditions of non-negativity, identity of indiscernibles, symmetry, and triangle inequality. It can be easily shown that both of the analyzed measures satisfy the first three conditions.

The satisfiability of the triangle inequality condition has been tested computationally. For this purpose, the three sets: *walking*, *swimming*, and *other* were merged into one set denoted as *w+s+o*. In order to test the influence of different components and their combinations on the triangle inequality violations, all possible combinations of binary weights: $w_V, w_D, w_N, w_G \in \{0, 1\}$ were tested. The results of the investigation on triangle inequality are presented in Table 1. It can be seen that the number of weight sets for which triangle inequality was violated is lower for $dissim_{FlexOpt}$. Also, the percentage of the cases for which violation occurred is significantly lower for $dissim_{FlexOpt}$ than for $dissim_{DegGreedy}$. For the cases in which only one component of the dissimilarity measure was taken into account, violations occurred only for the geometrical distance component using $dissim_{FlexOpt}$.

It is worth noting that for $dissim_{FlexOpt}$, the number of non-zero triangle inequality violations is the lowest when all four components of the dissimilarity measure are taken into account. These results suggest that the number of violations can be decreased by using all four components of the measure, i.e., by exploiting all the information about compared 3D structures the measure can access.

Computational cost. The results of time measurement for calculation of the full square distance matrix for different test sets are shown in Fig 5. In all cases, the calculation time for $dissim_{FlexOpt}$ is higher than for $dissim_{DegGreedy}$. The ratio

Test set	w_V	w_D	w_N	w_G	$dissim_{DegGreedy}$ [%]	$dissim_{FlexOpt}$ [%]
w+s+o	0	0	1	0	0.479	0.0
	0	0	0	1	0.158	0.004
	1	0	1	0	0.271	0.0
	1	0	0	1	0.154	0.0
	0	1	1	0	0.131	0.0
	0	1	0	1	0.123	0.010
	0	0	1	1	0.106	0.018
	1	1	0	1	0.115	0.021
	0	1	1	1	0.090	0.011
	1	1	1	0	0.130	0.0
	1	0	1	1	0.102	0.013
	1	1	1	1	0.089	0.003
best400	0	0	1	0	1.503	0.0
	0	0	0	1	2.604	0.810
	1	0	1	0	1.112	0.0
	1	0	0	1	2.549	0.807
	0	1	1	0	0.928	0.0
	0	1	0	1	2.352	0.794
	0	0	1	1	1.438	0.267
	1	1	0	1	2.566	0.788
	0	1	1	1	1.390	0.248
	1	1	1	0	0.927	0.0
	1	0	1	1	1.510	0.257
	1	1	1	1	1.277	0.246

Table 1: The percentage of triangle inequality violations for the original and the new dissimilarity measure and for different sets of weights. Combinations of weights for which there were no triangle inequality violations in both measures are not shown.

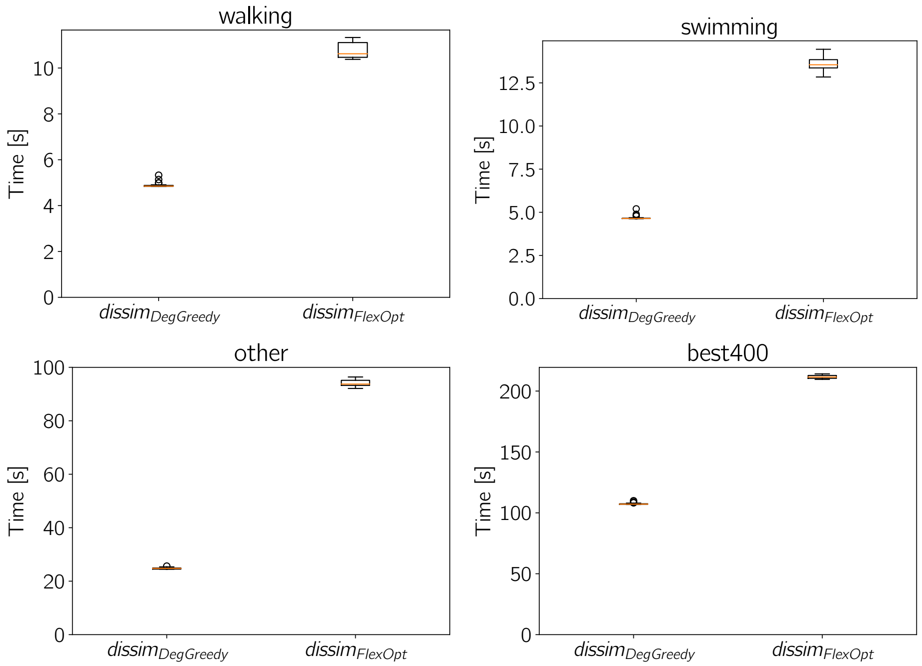


Fig. 5: Time of the full distance matrix calculation for each of the four test sets and both dissimilarity measures. All tests were performed on a computer with the Ubuntu Linux OS, the Intel Core i5-4200U processor and 4 GB of RAM.

of both calculation times is the highest for test sets containing structures with higher number of vertices – the calculation time for $dissim_{FlexOpt}$ is almost 4 times longer for *other*, and almost 3 times longer for *swimming* test set.

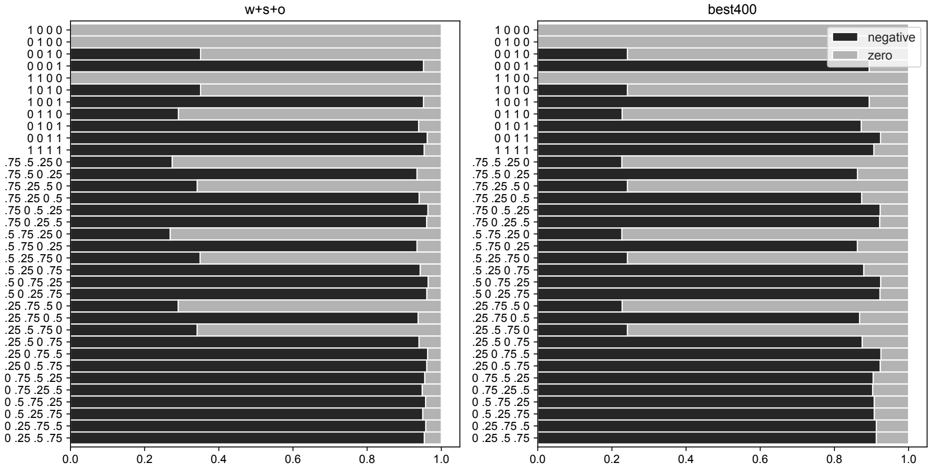


Fig. 6: The fraction of negative (black) and zero (gray) differences between the new and the previous measure for different test sets and different weight values w_V, w_D, w_N, and w_G. The differences were calculated as $dissim_{FlexOpt} - dissim_{DegGreedy}$.

Values of dissimilarity measures. Both measures were also compared in terms of the raw distance (dissimilarity) value – lower dissimilarity values suggest that the measure is able to find a better match between the structures. For this purpose, again, the three test sets: *walking*, *swimming* and *other* were merged into one set denoted as *w+s+o*. The number of negative, zero, and positive differences was calculated for distance matrices obtained using $dissim_{DegGreedy}$ and $dissim_{FlexOpt}$ with different weight combinations.

The results of this analysis are shown in Fig. 6. It can be seen that there were no positive differences, which means that distances obtained using $dissim_{FlexOpt}$ were always equal or lower than the distances obtained using $dissim_{DegGreedy}$. Distances obtained using both measures are the same only when the d_G and the d_N components are not taken into account. The number of negative differences is the highest when the d_G is included as a component of the measure. These results demonstrate that $dissim_{FlexOpt}$ can yield lower distance values then $dissim_{DegGreedy}$ in terms of the d_G and the d_N components, and such lower dissimilarities are the consequence of a better matching.

3 Fitness–distance correlation analysis

As mentioned in Sect. 1.1, the value of the fitness–distance correlation can be used to assess the global convexity of the fitness landscape. The results of the global convexity analysis can later be used to guide the design of efficient genetic representations and operators.

Both dissimilarity measures, $dissim_{DegGreedy}$ and $dissim_{FlexOpt}$, were used for fitness–distance correlation analysis, for different fitness functions, as shown in Fig. 7. Since the global optimum is not known and there are many local optima, mean distance to structures with the same or better fitness was computed, instead of the distance to the optimal structure. The analysis was conducted on four subsets of *best400*, each consisting of 100 structures evolved using one of the following maximized fitness functions:

- velocity on land (active structures: neural network activated),
- velocity in water (active structures: neural network activated),
- height (passive structures: no neural network),
- height (active structures: neural network activated).

For velocity on land, all the correlations were negative, meaning that the structures most similar to better or equally good structures (in terms of fitness) had in general higher fitness. This was the case for all three components, with the lowest correlation strength for d_N and the highest for d_D. This result suggests that velocity on land is correlated the most with the dissimilarity of the vertex degree.

For velocity in water, negative correlations were even stronger for d_G and d_N, showing correlation of dissimilarity in terms of these components with the fitness value. Surprisingly, for the d_N component, weak positive correlations were obtained.

Even more surprisingly, for maximizing the height of passive structures, high positive correlations were revealed for the d_G and the d_D components (since passive structures were considered, the d_N component was not taken into account). In this seemingly simple task, no global convexity was discovered. On the contrary, the more dissimilar the structure was to the better or equally fit structures, the higher was its fitness. This suggests the need to develop another dissimilarity measure (or another component of the measure) to help improve the efficiency of optimizing the height of passive structures. Such improvement would be possible if properties considered in the measure whose dissimilarity correlates with fitness were preserved by genetic operators that modify solutions during optimization.

For maximizing the height of active structures, moderate to high negative correlations were obtained. One difference is the correlation for the d_G component using $dissim_{DegGreedy}$, for which the correlation strength is near zero. Interestingly, correlation is significantly stronger for the d_G component using $dissim_{FlexOpt}$.

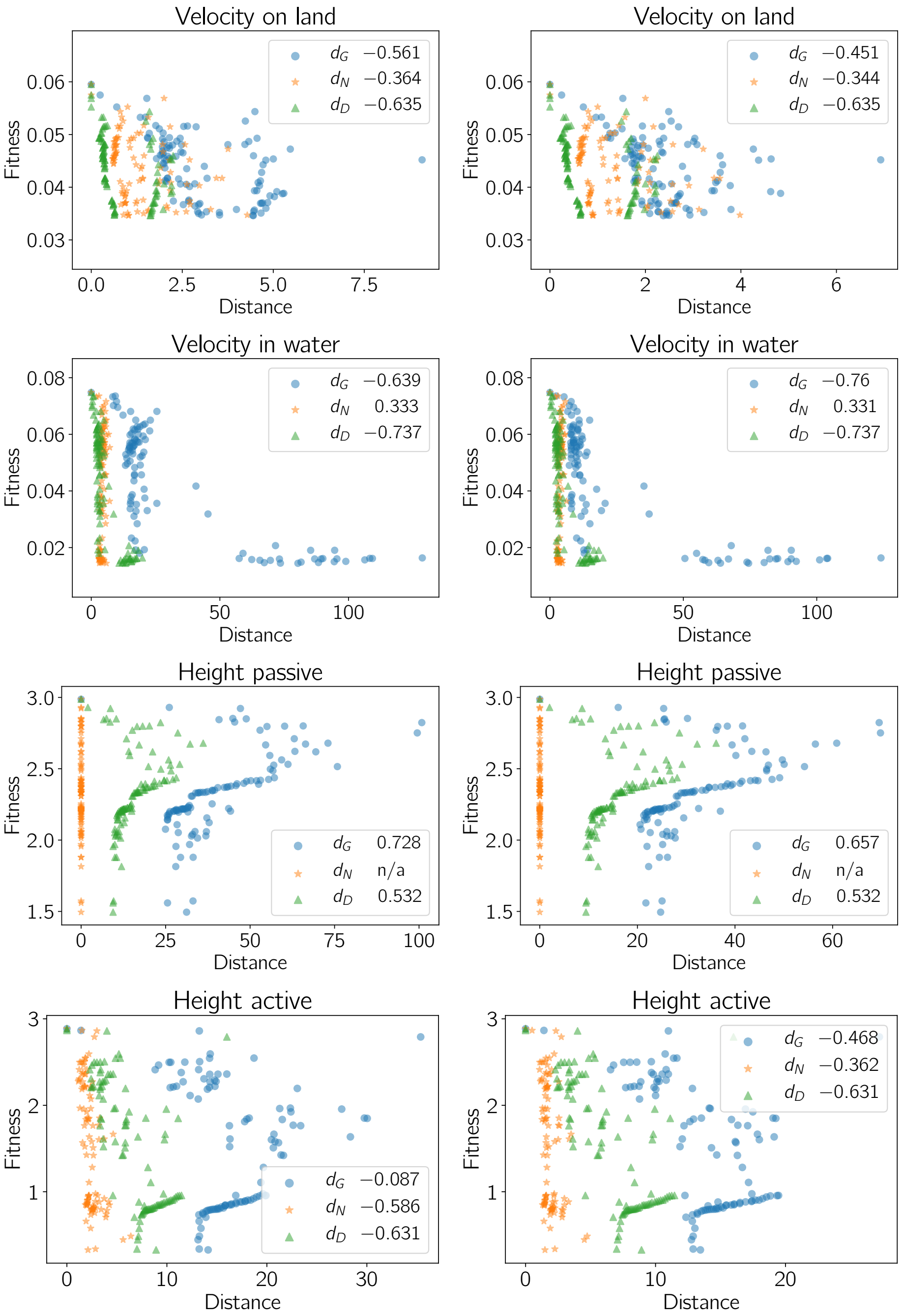


Fig. 7: The relationship between fitness value of the structure and a mean distance (dissimilarity) to structures with the same or better (i.e., higher) fitness value. Results for each component of the dissimilarity measure are presented separately on each plot, except for the d_V component, which is not taken into account. Spearman's r_s rank correlation coefficient values are shown in the legend for each of the components. Results obtained using $dissim_{DegGreedy}$ are shown in the left column, and results obtained using $dissim_{FlexOpt}$ are shown in the right column.

4 Conclusions and further work

The $dissim_{FlexOpt}$ measure introduced in this paper is an improvement over $dissim_{DegGreedy}$ in terms of the satisfiability of triangle inequality. It is not perfect, however – violations of this important property still occur, albeit very rarely. The measure can be further improved by eliminating the triangle inequality violation. Results of the analysis suggests that the geometrical distance component may be the main source of the violations. Therefore, the solution to this problem could be the change of the geometrical distance penalty for unmatched vertices. However, the penalty for the geometrical component is not so obvious as for the remaining components of dissimilarity measures. Currently, the penalty for the geometrical component is calculated as the distance of the unmatched vertex to the origin of the coordinate system; nonetheless, there are other possible approaches, some of which may fulfill the satisfiability of the triangle inequality condition.

In terms of the measure value, $dissim_{FlexOpt}$ introduced in this work is also an improvement over $dissim_{DegGreedy}$. The change of a greedy method of matching to the optimal matching lowered primarily the value of the geometrical distance component. Lower measure value indicates that $dissim_{FlexOpt}$ is able to find a better matching of both compared structures than $dissim_{DegGreedy}$. However, the price for those improvements is the increase in computational cost. While the execution time is still reasonable for sets containing structures with a moderate mean number of vertices, it may be prohibitive for data sets comprised of very complex structures.

Another advantage of the $dissim_{FlexOpt}$ measure is its flexibility. The earlier matching procedure based on the sorting of the vertices according to their vertex degree and neuron count was replaced by the Hungarian algorithm. Because of this change, all of the components of the dissimilarity measure are now processed uniformly. In consequence, the measure can take into account any property of the vertices and the influence of this property on the dissimilarity value will be proportional to the corresponding weight.

The fitness–distance correlation analysis revealed global convexity for most of the considered optimization tasks, as long as the vertex degree and the geometrical distance components were employed in the dissimilarity measure. While these results are promising, some unexpected relationships were discovered – like positive correlation between dissimilarity of the number of neurons and the velocity in water, or strong positive correlation between geometrical dissimilarity and the height of passive structures. These discoveries indicate that the development of other dissimilarity measures for 3D structures would be beneficial, as such measures would likely help identify properties of the optimized structures that yield high FDC. Incorporating such properties in the development of genetic representations and operators will in turn increase the efficiency of the optimization process in the area of evolutionary robotics and evolutionary design.

References

1. Hornby, G.S., Globus, A., Linden, D.S., Lohn, J.D.: Automated antenna design with evolutionary algorithms. In: AIAA Space. pp. 19–21 (2006)
2. Funes, P., Pollack, J.: Evolutionary body building: Adaptive physical designs for robots. Artificial Life 4(4), 337–357 (1998)
3. Byrne, J., Fenton, M., Hemberg, E., McDermott, J., O'Neill, M., Shotton, E., Nally, C.: Combining structural analysis and multi-objective criteria for evolutionary architectural design. Applications of Evolutionary Computation pp. 204–213 (2011)
4. Kicinger, R., Arciszewski, T., DeJong, K.: Evolutionary design of steel structures in tall buildings. Journal of Computing in Civil Engineering 19(3), 223–238 (2005)
5. Lipson, H., Pollack, J.B.: Automatic design and manufacture of robotic lifeforms. Nature 406(6799), 974–978 (2000)
6. Nolfi, S., Floreano, D.: Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines. MIT press (2000)
7. Corucci, F.: Evolutionary developmental soft robotics: towards adaptive and intelligent soft machines following nature's approach to design. In: Soft Robotics: Trends, Applications and Challenges, pp. 111–116. Springer (2017)
8. Sims, K.: Evolving 3D morphology and behavior by competition. Artificial life 1(4), 353–372 (1994)
9. Komosinski, M., Ulatowski, S.: Framsticks: Creating and Understanding Complexity of Life, chap. 5, pp. 107–148. Springer, New York, second edn. (2009), http://www.springer.com/978-1-84882-284-9
10. Komosinski, M.: Applications of a similarity measure in the analysis of populations of 3D agents. Journal of Computational Science (2017)
11. Bentley, P.J.: An introduction to evolutionary design by computers. In: Bentley, P.J. (ed.) Evolutionary Design by Computers. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edn. (1999)
12. Hornby, G.S., Pollack, J.B.: Generative representations for evolutionary design automation. Brandeis University (2003)
13. Smith, T., Husbands, P., O'Shea, M.: Fitness landscapes and evolvability. Evolutionary computation 10(1), 1–34 (2002)
14. He, J., Reeves, C., Witt, C., Yao, X.: A note on problem difficulty measures in black-box optimization: Classification, realizations and predictability. Evolutionary Computation 15(4), 435–443 (2007)
15. Boese, K.D.: Cost versus distance in the traveling salesman problem. UCLA Comp. Science Dept. (1995)
16. Pitzer, E., Affenzeller, M.: A comprehensive survey on fitness landscape analysis. In: Recent Advances in Intelligent Engineering Systems, pp. 161–191. Springer (2012)
17. Jones, T., Forrest, S.: Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In: Proceedings of the 6th International Conference on Genetic Algorithms. pp. 184–192. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1995)
18. Freisleben, B., Merz, P.: New genetic local search operators for the traveling salesman problem. Parallel Problem Solving from Nature – PPSN IV pp. 890–899 (1996)
19. Freisleben, B., Merz, P.: A genetic local search algorithm for solving symmetric and asymmetric traveling salesman problems. In: Proceedings of IEEE International Conference on Evolutionary Computation. pp. 616–621. IEEE (1996)

20. Jaszkiewicz, A., Kominek, P.: Genetic local search with distance preserving recombination operator for a vehicle routing problem. European Journal of Operational Research 151(2), 352–364 (2003)
21. Jaszkiewicz, A., Kominek, P., Kubiak, M.: Adaptation of the genetic local search algorithm to a car sequencing problem. In: 7th National Conference on Evolutionary Algorithms and Global Optimization, Kazimierz Dolny, Poland. pp. 67–74 (2004)
22. Kubiak, M.: Systematic construction of recombination operators for the vehicle routing problem. Foundations of Computing and Decision Sciences 29(3) (2004)
23. Cyr, C.M., Kimia, B.B.: 3D object recognition using shape similiarity-based aspect graph. In: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001. vol. 1, pp. 254–261 vol.1 (2001)
24. Barthel, D., Hirst, J.D., Błażewicz, J., Burke, E.K., Krasnogor, N.: ProCKSI: a decision support system for protein (structure) comparison, knowledge, similarity and information. BMC Bioinformatics 8(1), 416 (Oct 2007)
25. Bero, S.A., Muda, A.K., Choo, Y.H., Muda, N.A., Pratama, S.F.: Similarity measure for molecular structure: A brief review. Journal of Physics: Conference Series 892(1) (2017), http://stacks.iop.org/1742-6596/892/i=1/a=012015
26. Komosinski, M., Kubiak, M.: Quantitative measure of structural and geometric similarity of 3D morphologies. Complexity 16(6), 40–52 (2011)
27. Komosinski, M., Ulatowski, S.: Framsticks web site (2019), http://www.framsticks.com
28. Komosinski, M., Ulatowski, S.: Framsticks SDK (Software Development Kit), http://www.framsticks.com/sdk
29. Cox, T.F., Cox, M.A.A.: Multidimensional scaling. Chapman and Hall/CRC (2000)
30. Kuhn, H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly 2(1-2), 83–97 (1955)
31. Munkres, J.: Algorithms for the assignment and transportation problems. Journal of the society for industrial and applied mathematics 5(1), 32–38 (1957)