Journal of Memetic Computing manuscript No.
(will be inserted by the editor)

A genotype-phenotype-fitness Assessment Protocol for Evolutionary Self-Assembly Wang tiles Design

Germán Terrazas · Natalio Krasnogor

Received: date / Accepted: date

Abstract In a previous work we have reported on the evolutionary design optimisation of self-assembling Wang tiles capable of arranging themselves together into a target structure. Apart from the significant findings on how self-assembly is achieved, nothing has been yet said about the efficiency by which individuals were evolved. Specially in light that the mapping from genotype to phenotype and from this to fitness is clearly a complex, stochastic and non-linear relationship. One of the most common procedures would suggest running many experiments for different configurations followed by a fitness comparison, which is not only time-consuming but also inaccurate for such intricate mappings. In this paper we aim to report on a complementary dual assessment protocol to analyse whether our genetic algorithm, using morphological image analyses as fitness function, is an effective methodology. Thus, we present here fitness distance correlation to measure how effectively the fitness of an individual correlates to its genotypic distance to a known optimum, and introduce clustering as a mechanism to verify how the objective function can effectively differentiate between dissimilar phenotypes and classify similar ones for the purpose of selection.

G. Terrazas
Interdisciplinary Computing and Complex Systems (ICOS) Research Group
`http://icos.cs.nott.ac.uk`
School of Computer Science
University of Nottingham, UK
E-mail: German.Terrazas@nottingham.ac.uk

N. Krasnogor
Interdisciplinary Computing and Complex Systems (ICOS) Research Group
`http://icos.cs.nott.ac.uk`
School of Computer Science
University of Nottingham, UK
E-mail: Natalio.Krasnogor@nottingham.ac.uk

1 Introduction

Self-assembly systems are characterized by inorganic or living entities that achieve global order as the result of local interactions within a particular closed environment [16]. Self-assembly is a key cooperative mechanism in nature. Surprisingly, it has received (relatively) very little attention in computer science [28]. In [23], we defined the *self-assembly Wang tiles system* T_{sys} as a computational model of self-assembly. This system comprises a set of square tiles with labelled edges that randomly move across the Euclidean plane forming aggregates or bouncing off as result of their interaction (see Fig. 1). Cooperativity is an emergent feature of this system where the combination of a certain number of tiles is required to initiate self-assembly [17][21].

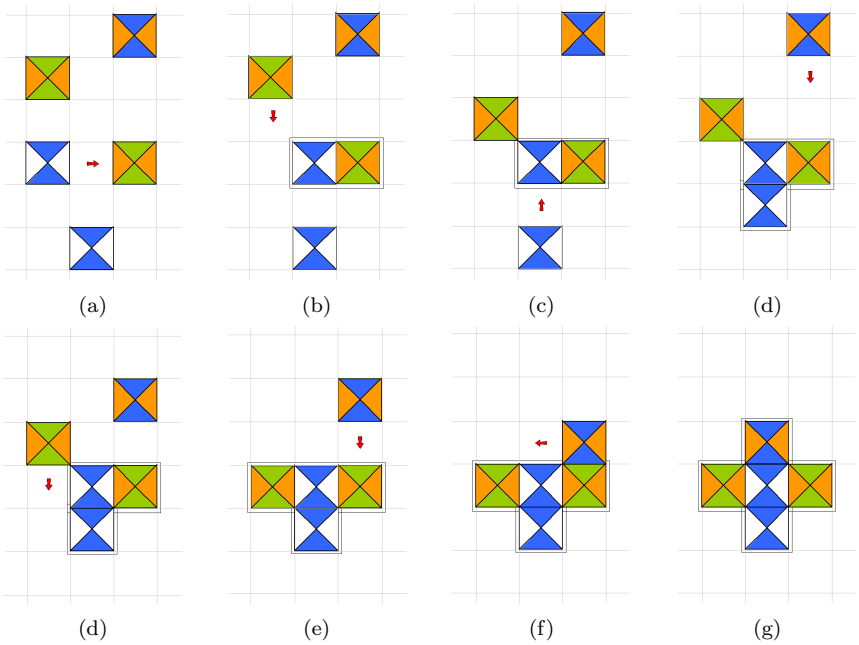


Fig. 1 A step-by-step (a to g) aggregate formed as the result of the interaction between five self-assembly Wang tiles performing random walk across a lattice.

Formally speaking, T_{sys} is defined as follows:

$$\begin{aligned} T_{sys} &= (\mathcal{T}, \Sigma, g, \mathcal{L}, \tau) \\ \mathcal{T} = \{t | t = (c_0, c_1, c_2, c_3)\} &\text{ where } c_0, c_1, c_2, c_3 \in \Sigma \\ g &:: \Sigma^2 \to \mathcal{Z}^+ \cup \{0\} \\ \tau &\in \mathbb{Z}^+ \cup \{0\} \end{aligned} \tag{1}$$

In this system, $\mathcal{T}$ is a non-empty finite set of Wang tile types t defined as a 4-tuple (c_0, c_1, c_2, c_3) which indicates the associated "glue" types labels at the

north, east, south and west edges of a tile, Σ is a set of symbols representing glue type labels, g is called the *glue function*, $\mathcal{L}$ is a lattice and τ is a threshold modelling the kinetic energy of the system. The glue function g is defined to compute the strength associated to a given pair of glue types. The lattice $\mathcal{L}$ is a two-dimensional surface with size $W \times H$ composed by a finite set of interconnected unit squared cells where tiles belonging to $\mathcal{T}$ are randomly located and perform random walks. Thus, when two or more tiles collide, the strength between the glue types at their colliding edges is calculated and compared to τ. If the resulting strength is bigger than τ, tiles self-assemble forming aggregates, otherwise they bounce off and keep moving.

Finding the appropriate combination of autonomous entities capable of arranging themselves together into a target configuration is a challenging open problem for the design and development of distributed cooperative systems. In [23] we addressed the self-assembly Wang tiles designability problem by means of artificial evolution. Our interest in combining self-assembly Wang tiles with evolutionary algorithms lays on the use of a method for the automated construction of supra-structures that emerge as the result of tiles interaction. In particular, we seek to answer the following:

Given a collective target configuration, is it possible to automatically design, e.g. with an evolutionary algorithm, the local interactions so they self-assemble into the desired target? That is, we want to design a set $\mathcal{T} = \{t|t = (c_0, c_1, c_2, c_3)\}$ *such that tiles of type* t *self-assemble into a given target shape.*

In order to address this question, our research was centred in the use of a genetic algorithm (GA) to evolve a population of self-assembly Wang tile families. Broadly speaking, a self-assembly Wang tile family is a descriptor comprising a set of glue types each of which is associated to one of the four sides of a self-assembly Wang tile. Thus, each tile family is instantiated with equal number of tiles which are randomly located into a simulation environment where they drift and interact one another self-assembling in aggregates until the simulation runs it course. Once the simulation finishes, aggregates are compared for similarity to a user defined (target) structure employing the Minkowski functionals [18] [19]. This assembly assessment returns a numerical representation that is considered as the fitness value (Fitness_i) of each individual. Thus, individuals capable of creating aggregates similar to the specified target are better ranked and become the most likely to survive across generations. This process, together with one-point crossover and bitwise mutation operators, is applied to the entire population and repeated for a certain number of generations. In particular, our experiments comprised four increasingly rich/complex simulation environments: deterministic, probabilistic, deterministic with rotation and probabilistic with rotation. An illustration summarising our approach and its components is shown in Fig. 2.

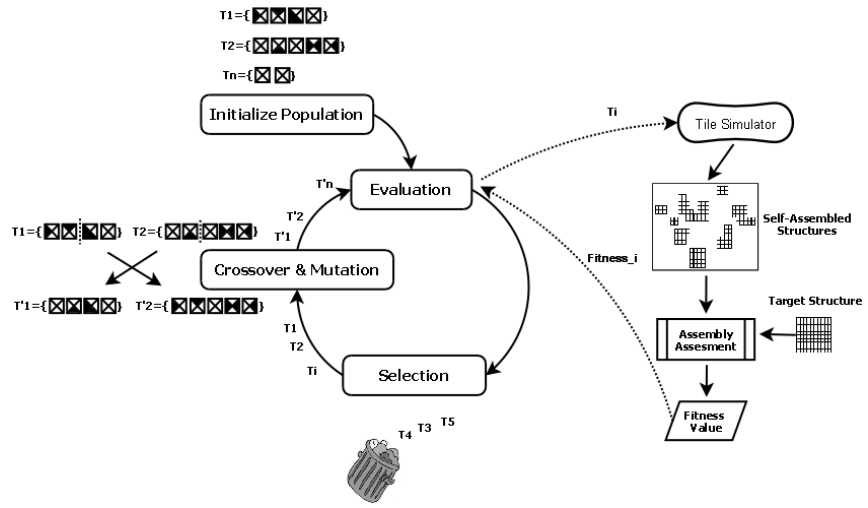


Fig. 2 Evolutionary approach for the evolutionary design optimisation of self-assembly Wang tiles. A population of self-assembling Wang tiles family (genotype) is randomly initialised. After that, each individual is set up into a tiles simulator from where the emerging self-assembled aggregations (phenotypes) are compared against a target structure for similarity. This comparison returns in the fitness of the individual. Later on, the application of genetic operators follows where the best ranked individuals are likely to pass throughout selection, crossover and mutation stages.

The achieved results supported our evolutionary design approach as a successful engineering mechanism for the computer-aided design of self-assembly Wang tiles. Early evidence of our research in this topic is available in [22] where we employed a very simple evaluation mechanism composed by a lattice scanner fitness function and later, in [23], where morphological image analyses brought a more accurate and efficient way to collectively assess the assembled aggregates towards the target. Since the mapping from genotype to phenotype and from this to fitness is clearly a complex, stochastic and non-linear relationship, would it be possible to analyse the effectiveness of Minkowski functionals as fitness function ? The aim of this paper is then to address this question with a complementary dual assessment protocol which tells us whether the employed genetic algorithm is an effective design optimisation method for our problem. Thus, in the following section we introduce fitness distance correlation and cluster analysis which set up the foundations of our novel protocol. Next we present how fitness distance correlation and cluster analysis are applied to our evolutionary design approach together with experiments and results. Finally, conclusions and discussions follow.

2 The Genotype-Phenotype-Fitness Assessment Protocol

The evolutionary design of self-assembly Wang tiles is characterised as a problem in which the mapping from genotype to phenotype and then from phe-

notype to fitness is a highly complex, non-linear and in some cases stochastic relationship. It is non-linear because different genotypes (tile sets) with small differences may lead to widely diverging phenotypes. While the same genotype, due to random effects, might produce a variety of end-products. This intricate relationship (see Fig. 3) makes the assessment of the genotype very difficult since the same (different) fitness value could be assigned to different (the same) genotypes. Hence, in order to analyse the efficiency by which individuals were evolved, we present in this section the fundamental components of a complementary dual assessment protocol: fitness distance correlation and cluster analysis. We have previously used this approach for the analysis of complex meta-cellular automata [24]. The former measures how effectively the fitness of an individual correlates to its genotypic distance to a known optimum. The latter verifies how the objective function can effectively differentiate between dissimilar phenotypes and classify similar ones for the purpose of selection.

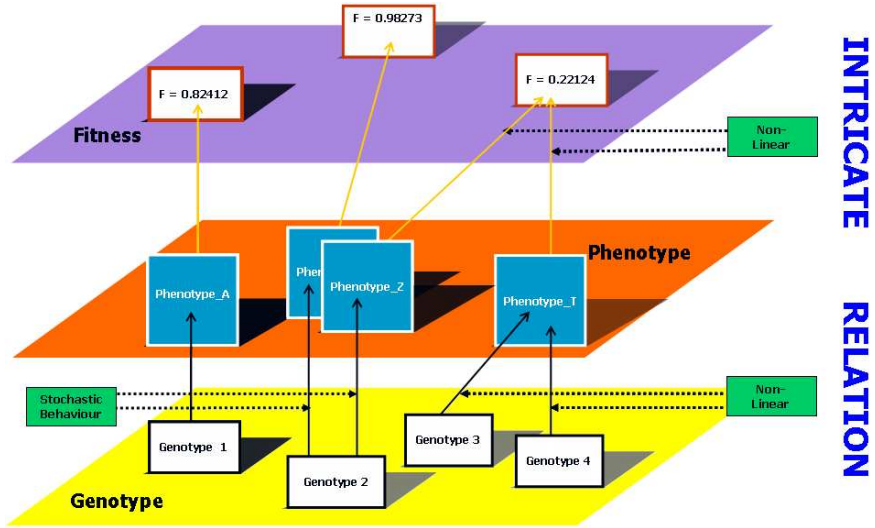


Fig. 3 The highly complex, non-linear and stochastic relationship taking place across the mapping from genotype to phenotype and then from phenotype to fitness.

2.1 Fitness Distance Correlation

Fitness Distance Correlation (FDC) [11] [10] is a summary statistic that performs a correlation analysis in terms of a known optimum and samples taken from the search space, predicting whether a GA will be effective for solving an optimisation problem. Thus, when facing a minimisation (maximisation) problem, a large positive (negative) correlation indicates that a GA may successfully treat the problem or that the problem is straightforward, whereas a

large negative (positive) value suggests that employing a GA may not be effective or that the problem is misleading. However, a correlation around zero, i.e. $-0.15 \leq \text{FDC} \leq 0.15$, would suggest that more nuisances, perhaps including scatter plots analyses, of the fitness versus distance to the optimum should be done and, in general the problem is categorized as difficult. The formula for computing the FDC is shown in Equation 2, where n is the number of samples, f_i is the fitness of sample i with distance to the known optimum d_i, $\overline{f}$ and S_F are the mean and standard deviation of the fitness values, and $\overline{d}$ and S_D are the mean and standard deviation of the distances.

$$
\begin{aligned}
FDC &= \frac{(1/n)\sum_{i=1}^{n}(f_i - \overline{f})(d_i - \overline{d})}{S_F S_D} \\
\overline{f} &= \frac{1}{n}\sum_{i=1}^{n} f_i \\
\overline{d} &= \frac{1}{n}\sum_{i=1}^{n} d_i \\
S_F &= \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(f_i - \overline{f})^2} \\
S_D &= \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(d_i - \overline{d})^2}
\end{aligned}
\tag{2}
$$

A study focused on whether FDC predicts the GA behaviour, and whether it detects differences in encoding and representation for a number of well-studied minimisation problems has been given in [11]. When predicting the GA behaviour, the FDC confirmed that Deb & Goldberg's 6-bit fully deceptive function and Whitley's 4-bit (F2 and F3) fully deceptive functions are indeed misleading since the correlation values were 0.30, 0.51 and 0.36 respectively, and the fitnesses tended to increase with the distance from the global optimum. In addition, FDC also confirmed that problems like Ackley's One Max, Two Max, Porcupine and NK landscape problems for $\text{K} \leq 3$ are easy since the correlation values resulted in -0.83, -0.55, -0.88 and -0.35 respectively. Nevertheless, the FDC indicated that NK(12,11) landscape, Walsh polynomials on 32 bits with 32 terms each of order 8, Royal Road functions R1 and R2, as well as some of the De Jong's functions like F2(12) are difficult since the resulting correlation values were close to 0.0. When the differences in encoding and representation were considered, experiments using Gray and binary coding led to the conclusions that the superiority depends on the number of bytes used to encode the numeric values. For instance, De Jong's F2 with binary coding is likely to make the search easier than with Gray coding when using 8 bits. In contrast, the correlation value of F12 indicated that Gray coding works better than binary when using 12 or 24 bits. Despite its successful application on a wide benchmarking set of problems, FDC is still not considered to be a very accurate predictor in some other problems. For

instance, a case where FDC failed as a difficulty predictor has been presented when studying a GA maximising *Ridge functions* [20].

In summary, although FDC cannot be expected to be a perfect predictor of performance, previous work reported in [13] [1] [26] [25] [27] suggests that it is indeed a good *indicator* of performance. Our goal is then to assess how effectively the fitness of an individual correlates to its genotype when using Minkowski functionals as fitness function for the GA presented in [23].

2.2 Cluster Analysis

One of the characteristics of our problem is non-linearity, i.e. several different genotypes can encode the same phenotype and hence introduce severe noise in the FDC analysis. As, ultimately, selection is based on fitness which in turn depends on the phenotype, studying the phenotype-fitness mapping could shed light on why the GA worked quite well. For this reason, this section introduces clustering as a method for analysing the phenotype-fitness relationship in self-assembly Wang tiles design optimisation.

Cluster analysis or clustering is a technique for grouping objects according to their similarities [6] [9] [3] [2]. In contrast to classification, clustering is an unsupervised task in which a set of objects is partitioned in groups, called clusters, according to their proximities such that those belonging to a cluster are more similar to each other than objects in a different cluster. A clustering procedure comprises four basic stages: feature selection or extraction, clustering algorithm selection, cluster validation and results interpretation.
In the first stage, the features by which the objects will be distinguished are chosen. This pairwise affinity is then considered to compute a proximity matrix to which a cluster strategy is applied. The resulting partition of the data is subject to a subsequent testing criteria in order to validate the clustering process. Finally, a visualization and interpretation over the achieved clusters closes the procedure with the hope of providing meaningful insights coming from the original data. The whole inter-relation among these tasks can be seen as a sequential procedure as detailed and commented in Fig. 4.

Clustering algorithms are classified as hierarchical, partitioning, density-based partitioning, grid-based, evolutionary methods and so forth in [2] [14] [7]. Although there are slight differences among the proposed taxonomies, many are the common features associated with them [9]. For instance, according to their structure and operation, a clustering algorithm is agglomerative if clusters arise from singletons (bottom-up) or divisive if one super cluster is split in several ones (top-down). The sequential or simultaneous use of object features in the clustering process also plays an important role as it defines whether the algorithm is monothetic or polythetic. Additionally, it is also reported that some methodologies allow objects to belong to a single cluster (hard classification) or to multiple clusters (fuzzy classification). Besides these three characterizations, a clustering algorithm can also be deterministic, stochastic,

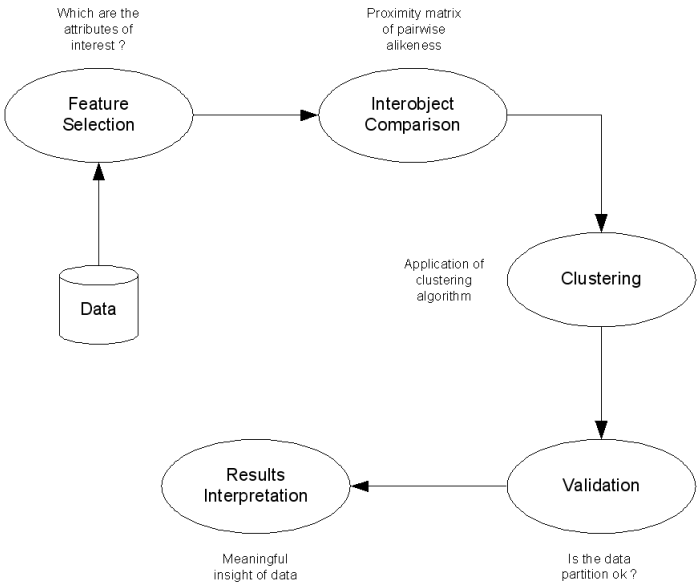


Fig. 4 Clustering procedure comprising feature selection, inter-object comparison, clustering, validation of data partition and results interpretation.

incremental or non-incremental if there are constraints on execution time or memory affecting the architecture of the algorithm.

The clustering methods appearing in the literature are, mainly, variants of the hierarchical agglomerative clustering. Among them, the single-link (SLINK), complete-link (CLINK) or minimum-variance [12] are the best-known where their differences lay on the way they characterise the similarity between pairs of clusters. In addition, the UPGMA, Weighted Pair Group Method using arithmetic Average (WPGMA), the Unweighted Pair Group Method using Centroids (UPGMC) and the Weighted Pair Group Method using Centroids (WPGMC) are also broadly employed in many applications [5] [8] [15].

3 Correlating the Self-Assembly Wang Tiles Design

Since different set of tiles may self-assemble in aggregates similar in shape to the target structure, it is of our interest to study here how effectively the fitness of an individual correlates to its genotypic distance to a known optimum (see Fig. 5). In the rest of this section, we carry out FDC analysis in order to study Minkowski functionals effectiveness as fitness function for the evaluation of the achieved self-assembled aggregates.

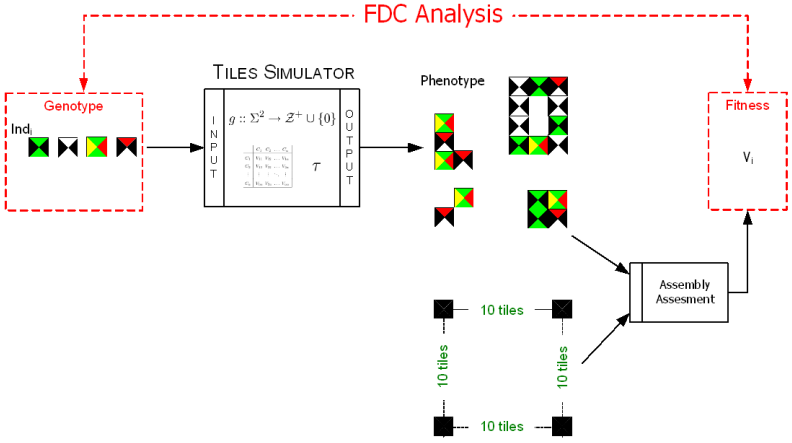


Fig. 5 Diagram of mappings from genotype onto phenotype and from phenotype onto numerical fitness value, and relationship to the Fitness Distance Correlation.

3.1 Experiments and Results

In order to perform a FDC analysis, we first choose the best individual found among the four simulation environments. In this case, the best individual belongs to the results achieved when using probabilistic criteria and no rotation simulator. Next, a data set comprising 500 individuals at different Hamming distances from the best individual is created. In particular, given two individuals Ind_i and Ind_j of same length, their Hamming distance H is defined as in Equation 3.

$$
\begin{aligned}
H(Ind_i, Ind_j) &= \sum_{k=1}^{n} diff(T_k^i, T_k^j) \\
diff(T_i, T_j) &= \sum_{l=0}^{3} c_l^i \ominus c_l^j \\
c_a \ominus c_b &= \begin{cases} 1 \text{ if } c_a \neq c_b \\ 0 \text{ otherwise} \end{cases} \\
& \quad c_a, c_b \in \Sigma
\end{aligned} \tag{3}
$$

Thus, this 500 individuals data set comprises all the possible chromosomes at Hamming distance of 1 plus some other randomly generated individuals at greater distance, all of these systematically generated following the pseudocode described in Algorithm 1 where $DuplicateReplacing(T_k, c_l, c_{new})$ duplicates tile T_k replacing glue type c_l with c_{new}, $DuplicateReplacing(Ind_i, T_k, T_{new})$ duplicates individual Ind_i replacing tile T_k with T_{new}, $TileAt(Ind_i, k)$ returns the tile at position k of an individual, and $Replace(T_k, c_l, c_{new})$ replaces glue type c_l in tile T_k with glue type c_{new}.

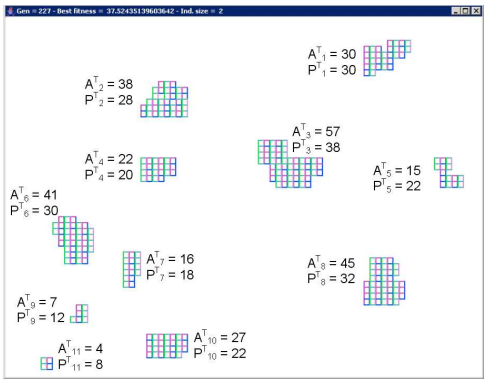
Algorithm 1 GenerateIndividuals

Require: Ind an individual
Ensure: S a set of individuals
1. **for all** tiles T_k in Ind_i **do**
2. **for all** glue types c_l in T_k **do**
3. **for all** glue type $c_{new} \in \Sigma$ **do**
4. $T_{new} \leftarrow DuplicateReplacing(T_k, c_l, c_{new})$
5. $Ind_{new} \leftarrow DuplicateReplacing(Ind_i, T_k, T_{new})$
6. $Insert(S, Ind_{new})$
7. **end for**
8. **end for**
9. **end for**
10. **while** $\mid S \mid < 500$ **do**
11. $Ind_{new} \leftarrow Duplicate$(randomly chosen $Ind_i \in S$)
12. $n \leftarrow Random(0, \mid Ind_{new} \mid)$
13. **for all** k to n **do**
14. $T_k \leftarrow TileAt(Ind_{new}, k)$
15. $m \leftarrow Random(0, 3)$
16. **for all** l to m **do**
17. $c_{new} \leftarrow Random(\Sigma \setminus c_l)$
18. $Replace(T_k, c_l, c_{new})$
19. $Insert(S, Ind_{new})$
20. **end for**
21. **end for**
22. **end while**

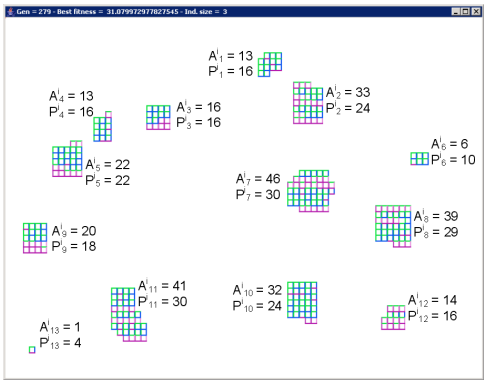
In total, each of the generated individuals was simulated 5 times giving as a result a group with equal number of final configurations. Thus, a configuration in turn was considered as a target ($Conf_T$) against which the remaining configurations of all the groups ($Conf_i$) were evaluated on fitness (f_i) and on distance (d_i) among their associate genotypes (see Equation 4).

$$\begin{aligned} d_i &= H(ind_i, ind_T) \\ f_i = f(Conf_i) = Eval(Conf_i, Conf_T) &= \sqrt{(\Delta\mathcal{A})^2 + (\Delta\mathcal{P})^2 + (\Delta\mathcal{N})^2} \\ \Delta\mathcal{A} &= max\{A_1^T, \ldots, A_m^T\} - max\{A_1^i, \ldots, A_n^i\} \\ \Delta\mathcal{P} &= \textstyle\sum_{k=1}^{m} P_k^T - \sum_{k=1}^{n} P_k^i \\ \Delta\mathcal{N} &= m - n \end{aligned} \quad (4)$$

Since a configuration comprises a collection of aggregates, a way is needed to perform an evaluation involving all its aggregations collectively. For this reason, considering a target configuration $Conf_T$ and an arbitrary one $Conf_i$ with aggregates $\{A_1^T, \ldots, A_m^T\}$ and $\{A_1^i, \ldots, A_n^i\}$ respectively, $Conf_i$ will be evaluated upon $Conf_T$ in terms of the difference in areas, perimeters and number of achieved aggregations as shown in Equation 4. In order to illustrate an example, Fig. 6 shows both a target configuration $Conf_T$ and an arbitrary configuration $Conf_i$, the fitness of which is calculated in terms of areas and perimeters in Equation 5.



$(Conf_T)$



$(Conf_i)$

Fig. 6 A target configuration ($Conf_T$) towards which the fitness of an arbitrary configuration ($Conf_i$) is calculated in terms of areas and perimeters. Tags A_j^T, P_j^T and A_k^i, P_k^i label area and perimeter values for aggregates in $Conf_T$ and $Conf_i$ configurations respectively.

$$\begin{aligned} \Delta\mathcal{A} &= max\{A_1^T, \ldots, A_13^T\} - max\{A_1^i, \ldots, A_{11}^i\} = 57 - 46 = 11 \\ \Delta\mathcal{P} &= \textstyle\sum_{k=1}^{13} P_k^T - \sum_{k=1}^{11} P_k^i = 260 - 255 = 5 \\ \Delta\mathcal{N} &= m - n = 11 - 13 = -2 \\ f_i &= \sqrt{(\Delta\mathcal{A})^2 + (\Delta\mathcal{P})^2 + (\Delta\mathcal{N})^2} = \sqrt{11^2 + 5^2 + (-2)^2} = 12.24745 \end{aligned} \quad (5)$$

After performing the calculations, the findings show that the FDC values range from -0.331444 to 0.281457. Since Equation 4 defines a minimisation, 50.60% of the FDC values indicate that using a GA may not be effective, 44.72% that the problem is difficult to solve and a 4.68% that the GA may successfully treat the problem (see Fig. 7).

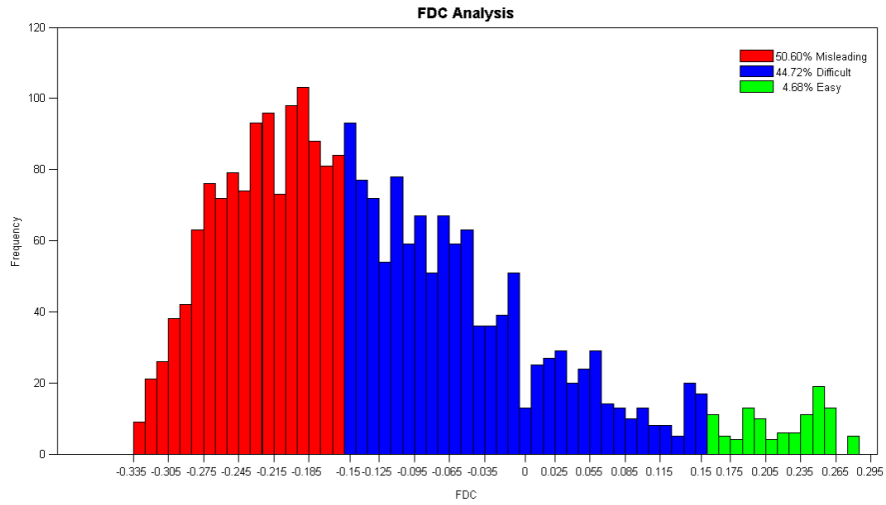


Fig. 7 Proportion of FDC values falling into difficult, misleading and easy to solve categories. From the 2500 analyses performed over 500 individuals, only a 4.68% reveals that a GA may successfully treat the problem.

In particular, visual inspections over scatter plots obtained from the values captured into the smallest percentage depict good correlation on some individuals. A representative sample of these is depicted in Fig. 8 but at `http://www.cs.nott.ac.uk/~gzt/fdcminkowski` we provide the rest of the experiments. For each plot, ind_ij identifies the j-simulation of individual i, where $j \in \{a, b, c, d, e\}$ and $i \in \{1, \ldots, 500\}$. Hence, from the sampling of 500 individuals and 2500 simulations subject to FDC analyses, it emerges that employing Minkowski functionals as fitness function offers a relatively satisfactory correlation upon the relationship genotype-fitness only for half of the putative samples.

Contrary to the interpretations given by some of the FDC figures seen in this section, the results reported in [23] reveal that using Minkowsi functionals as evaluation method of a GA has positively addressed the self-assembly Wang tiles design problem. Henceforth, analyses on the phenotype-fitness relationship are expected to shed light on the reasons for which such evolutionary approach has been effective.

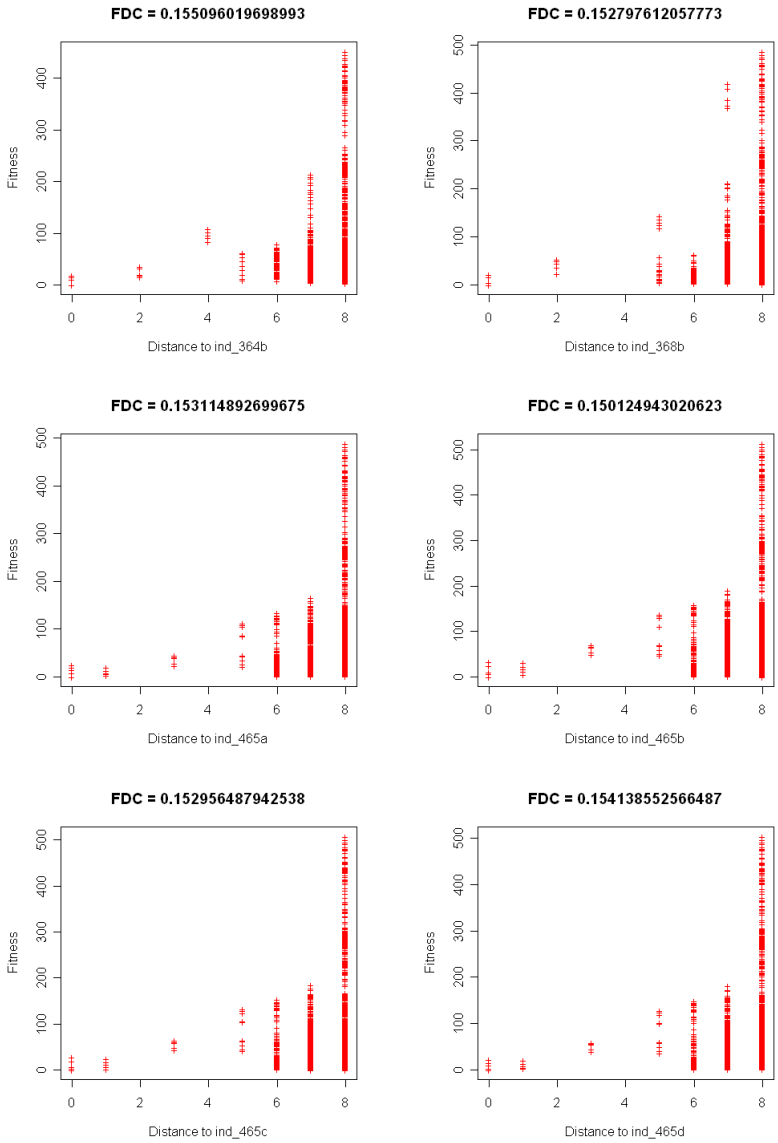


Fig. 8 Graphics of the resultant scatter plots and correlation coefficients for the self-assembly Wang tiles model showing that the Minkowski functionals has a relatively satisfactory correlation with the genotype for some of the self-assembly Wang tile families.

4 Clustering the Self-Assembly Wang Tiles Design

In this section, a clustering procedure will be applied at phenotype level with the hope of obtaining a better insight and of verifying the phenotype-fitness relationship (see Fig. 9). For this to be effective, the cluster analysis is considered over the final configurations obtained by the individuals created with the algorithm of Section 3.1 and from where the resulting findings of this methodology will be used as a complementary assessment of the Minkowski functionals as fitness function.

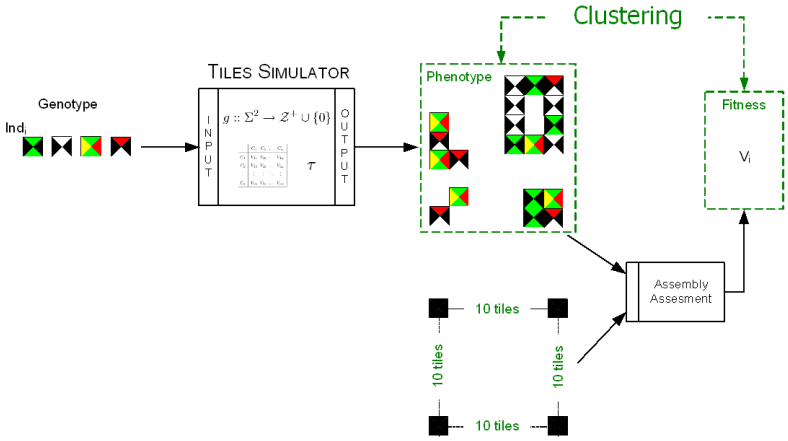


Fig. 9 Diagram of mappings from genotype onto phenotype and from phenotype onto numerical fitness value, and relationship to clustering analysis.

4.1 Experiments and Results

In order to perform the clustering experiments, the whole set of final configurations obtained from the self-assembly Wang tile simulations performed in Section 3.1 is employed. As these configurations comprise the actual clustering input data, the feature selection by which the objects are expected to be distinguished is the number of aggregates, their perimeters and the biggest aggregate area. Thereby, the measure of affinity between each pair of configurations $(Conf_i, Conf_j)$ is computed in terms of the evaluation function (see Equation 4) and stored in the similarity matrix M_s defined in Equation 6.

$$\begin{aligned} M_s[i,j] &= \begin{cases} Eval(Conf_i, Conf_j) & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases} \\ M_s &\text{ is a proximity matrix s.t. } M_s[i,j] = M_s[j,i] \\ Conf_i&,\ Conf_j \text{ are final configurations} \\ 1 &\leq i,j \leq 2500 \end{aligned} \tag{6}$$

Although a number of different clustering methods and representations are available, the unweighted pair-group method using arithmetic average (UPGMA) [4] has been chosen along with a logarithm dendogram representation to visualize and interpret the data partition. The pseudocode for the clustering algorithm UPGMA is outlined in Algorithm 2 where $MergeRows$ joins the content of row i and row j, $MergeColumns$ joins the content of column i and column j and $MakeNode$ associates i and j in a node that $InsertNode$ will add to the hierarchical structure T.

Algorithm 2 MakeCluster

Require: M_s a proximity symmetric matrix
Ensure: T a hierarchical structure
1. **while** $dimension(M_s) \geq 3 \times 3$ **do**
2. **for all** i, j such that $i \neq j$ **do**
3. $minimum \leftarrow min(M_s[i,j], Minimum)$
4. **end for**
5. $MergeRows(M_s, i, j)$
6. $MergeColumns(M_s, i, j)$
7. $M_s[k, ij] \leftarrow avg(M_s[k,i], M_s[k,j])$
8. $M_s[ij, k] \leftarrow avg(M_s[i,k], M_s[j,k])$
9. $node \leftarrow MakeNode(i, j)$
10. $InsertNode(T, node, minimum)$
11. **end while**

Thus, the resultant data partition showing eight clusters labelled as **A**, **B**, **C**, **D**, **E**, **F**, **G** and **H** is depicted in Fig. 10. By sampling representative configurations from each of these clusters, it is possible to observe that the data has been well partitioned since the distribution and morphology of the aggregations is mostly similar in each of the clusters.

On the one hand, by analysing the partitions located at the top part of the dendogram, the configurations of cluster **H** reveal scattered tiles with no or very few small aggregates like those appearing in the snapshot of Fig. 11 (a). Conversely, representatives of cluster **G** reveal large-size aggregates with very few unassembled tiles as in the configuration depicted in Fig. 11 (b). Close to these two types of partitions, the configurations of cluster **F** have achieved assemblies which are either large in size and merged with some small others as shown in Fig. 11 (c), or medium-size aggregates combined with few others of minor area as shown in the configuration in Fig. 11 (d).

On the other hand, from an analysis of some representatives belonging to the bottom right partitions labelled as **A** and **B**, it is evident that the morphology of the aggregates is contrary to the ones described before. For instance, the configurations observed in **A** comprise either aggregates with dendritic shape along with scattered tiles as shown in Fig. 12 (a), or small strips also sharing the lattice with few unconnected tiles as depicted in Fig. 12 (b). Similarly, the aggregations found in cluster **B** also comprise small rectangular aggregates

although mixed with T-shaped and L-shaped structures in most of the cases as appears in configurations of Fig. 12 (c) and Fig. 12 (d).

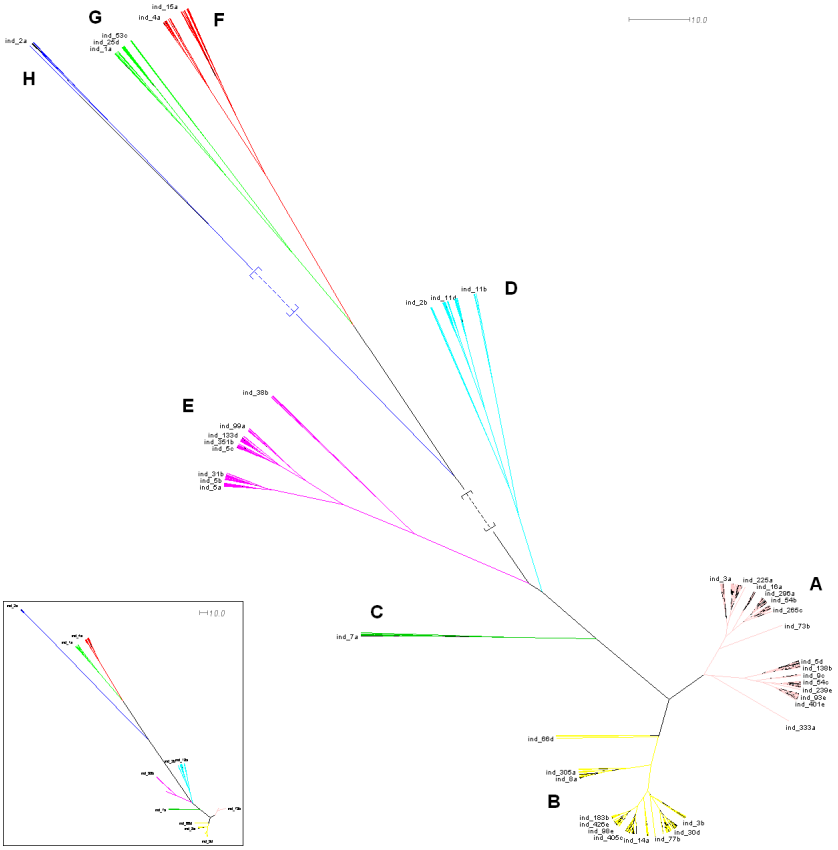
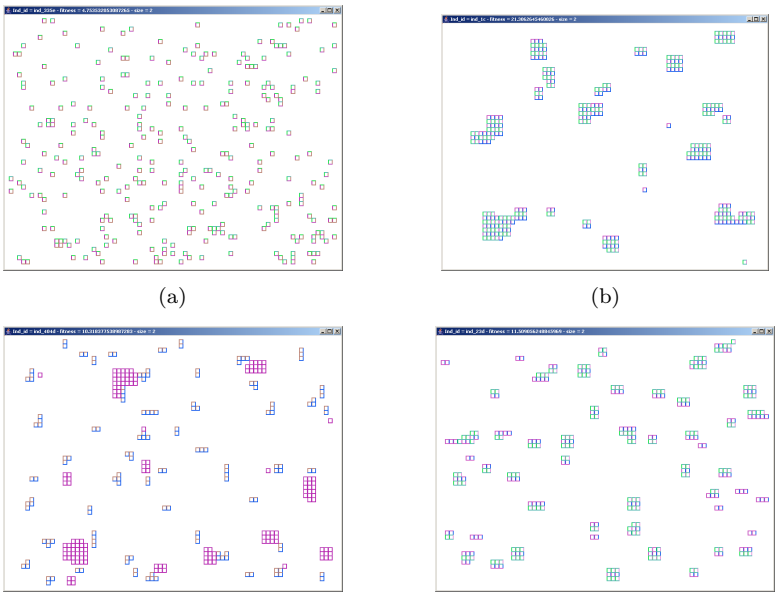


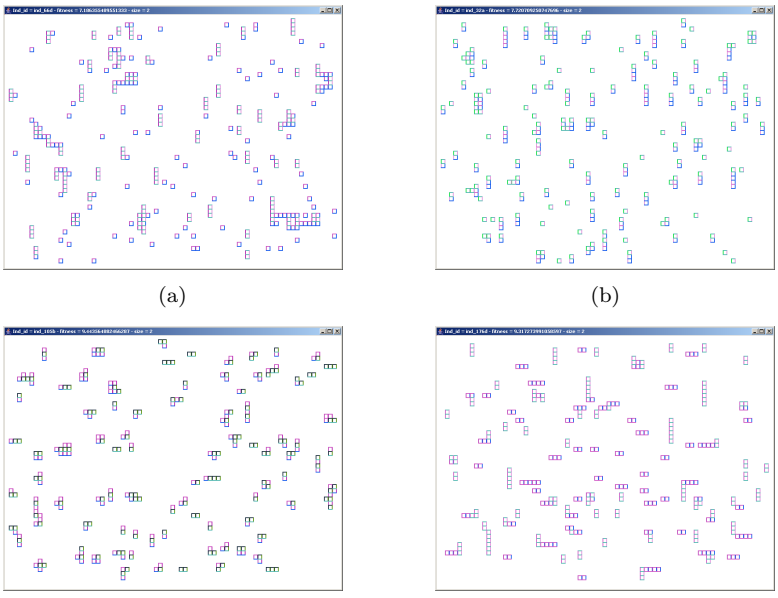
Fig. 10 Illustration of the logarithmic cluster tree for self-assembly Wang tiles configurations, individuals of which were obtained with Algorithm 1.

The three partitions located at the central part of the dendogram seem to represent a transition between the two analyses done above. For instance, the configurations belonging to cluster **D** mostly show medium-size strips together with a vast number of scattered tiles distributed across the lattice as it is shown in Fig. 13 (a) and Fig. 13 (b). Configurations with similar morphology and a reduced number of scattered tiles are among those observed in cluster **E** (see Fig. 13 (c)). Moreover, the same partition also includes some other type of configurations where aggregates are usually large and, in many cases, surrounded by scattered tiles (see Fig. 13 (d)). Finally, the occurrence of large- and medium-size aggregates combined with scattered tiles is the common feature that identifies the configurations observed in partition **C** (see Fig. 14).



(a) (b) (c) (d)

Fig. 11 Representatives of three clusters: (a) Scattered tiles and small size aggregates characterise partition **H**; (b) large aggregates feature partition **G**; (c-d) large and small size aggregates as well as medium and small size aggregates characterise partition **F**.



(a) (b) (c) (d)

Fig. 12 Representatives of two clusters: (a-b) dendritic aggregates along with scattered tiles and small strips with few unconnected tiles characterise partition **A**; (c-d) the appearance of T-shaped and L-shaped structures characterise partition **B**.

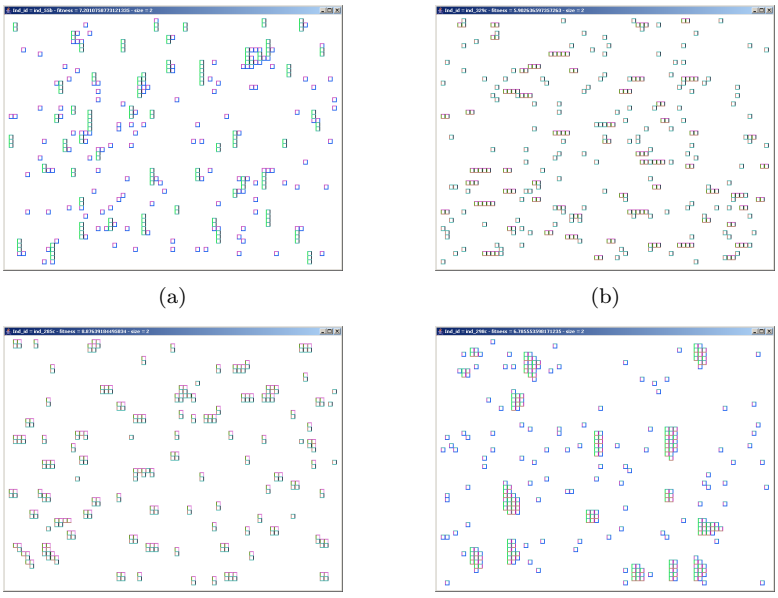


Fig. 13 Representatives of two clusters: (a-b) medium strips surrounded by a vast number of scattered tiles distributed across the lattice identify partition **D**; (c-d) a number of scattered tiles approaching to nil and some big aggregates characterise partition **E**.

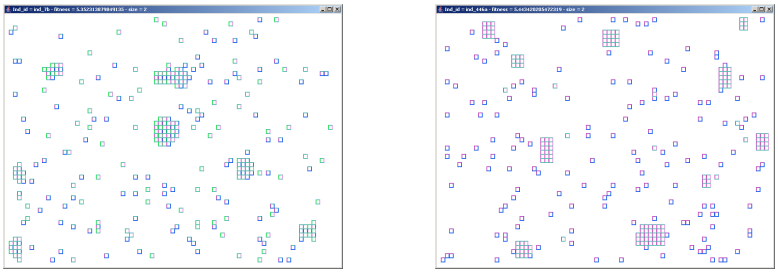


Fig. 14 Two representatives configurations of partition **C** showing large- and medium-size aggregates combined with unassembled tiles distributed across the lattice.

To summarise, the findings achieved after applying cluster analyses over the 2500 final configurations, i.e. product of simulating 500 individuals, come to support the view that there is in fact an acceptably high correlation between the phenotypes and their fitness values. In other words, these findings verify that Minkowski functionals can effectively differentiate between dissimilar phenotypes and classify similar ones for the purpose of selection.

5 Summary

As a general conclusion, the combination of FDC plus cluster analysis indicates that the use of Minkowski functionals is amenable for the evolutionary design optimisation of self-assembly Wang tiles. On the one hand, the results obtained with the morphological image analyses supports the use of Minkowski functionals as fitness function, although only 5% of the FDC analysis applied to the systematically obtained individuals has revealed that the use of GA may successfully solve the problem. On the other hand, the cluster analyses have accurately classified the configurations according to their morphological features, supporting the way in which the fitness function evaluates the self-assembled aggregates.

Hence, the application of our dual methodology, shown along Section 3 and Section 4, reveals that employing a fitness function in terms of Minkowski functionals for the evolutionary design optimisation of self-assembly Wang tiles results in a complex mechanism of evaluation where, although its success as phenotype evaluator seems to be appropriate, a different type of analysis is needed for an assessment of how effectively an individual correlates to its genotypic distance to a known optimum.

Considering the results presented here, we can conclude that employing the combination clustering and FDC is a dual assessment that reveals an accurate indication of the quality of the encoding, i.e. genotype, its mapping to phenotype and Minkowski functionals as fitness function. Therefore, the application of this methodology before starting long and expensive evolutionary runs should be considered in any problem where the genotype-phenotype-fitness mapping is complex, stochastic, many-to-many and computationally expensive. Thus, this protocol analysis is a contribution of general interest beyond automated self-assembly design.

6 Conclusions and Future Work

This paper has presented a dual assessment protocol to study the effectiveness of a GA employed as method for the design optimisation of self-assembly Wang tiles. Such is the complexity of the genotype-phenotype-fitness mapping, that FDC cannot, alone, be guaranteed to give a completely accurate picture. Indeed, the objective function itself is also only an approximation of two individuals' phenotypic similarity. For these reasons, relying on only FDC or only clustering to validate complex problems would not be adequate. It is for this reason that we combined both methods with the aim to show whether a given fitness function is a suitable evaluation mechanism for the evolutionary design problem addressed in [23].

Overall, we have contributed with a complementary dual assessment for validating complex, stochastic, non-linear genotype-phetontype-fitness relationships. The proposed methodology combines FDC and clustering validation techniques, the results of which revealed that Minkowski functionals are

suitable fitness functions for the evaluation of self-assembled structures. In addition, these indicators have contributed with a more general result which is a method for assessing the quality of the encoding and the accuracy of its mapping to phenotype in evolutionary systems where genotype-phenotype-fitness is an intricate relationship.

Self-assembly can be seen as an information-driven process and hence be better exploited by directly linking it to computation. Taken as an operational hypothesis this assumption implies that desired emergent phenomena could in principle be programmed into self-assembling nanosystems. Thus, we are currently focused on employing self-assembly Wang tiles model for the design and exploitation of molecular self-assembly, in particular porphyrins-based nanotiles. Porphyrins are suitable molecules since they have four fold symmetry, could be synthesized with different functional groups, and are planars, ideal for surface deposition on solid substrates see Fig. 15. In this way, given a set of self-assembly Wang tiles that self-assemble into specific patterns, our goal is to manufacture porphyrin molecules which would be able to self-assemble into similar structures by exploiting preferential intermolecular values.

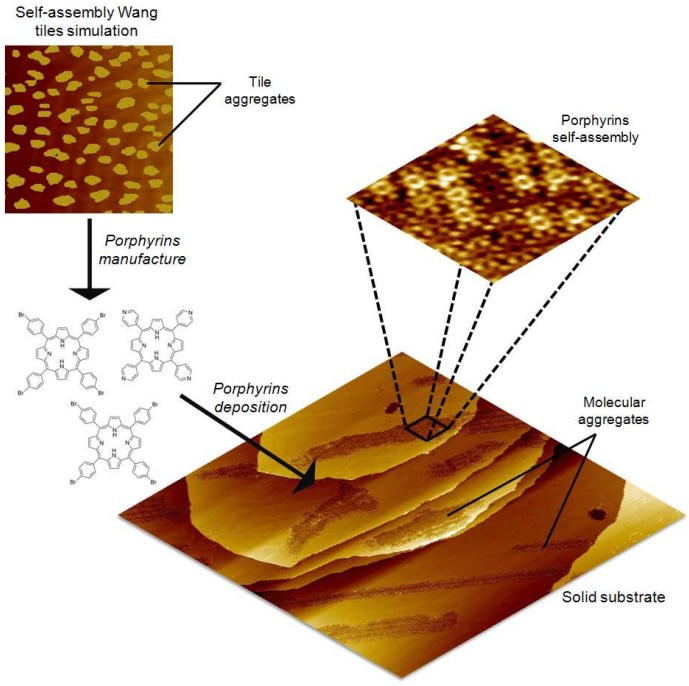


Fig. 15 Design and exploitation of porphyrins-based nanotiles. Sself-assembly Wang tiles that create aggregates are seen as blueprints for the manufacture of functionalised porphyrins. These molecules would then be deposited on a solid substrate where intermolecular binding strengths give origin to porphyrins self-assembly.

Acknowledgements The research reported here is funded by EPSRC grant EP/H010432/1 Evolutionary Optimisation of Self Assembling Nano-Designs (ExIStENcE) and a Leadership Fellowship (Natalio Krasnogor) EP/J004111/1.

References

1. Altenberg L (1997) Fitness Distance Correlation Analysis: An Instructive Counterexample. In: 7th International Conference on Genetic Algorithms, Morgan Kaufmann, San Francisco, CA, USA, pp 57–64
2. Berkhin P (2002) Survey of Clustering Data Mining Techniques. Tech. rep., Accrue Software, San Jose, CA, USA
3. Berkhin P (2006) A Survey of Clustering Data Mining Techniques. In: Kogan J, Nicholas C, Teboulle M (eds) Grouping Multidimensional Data, Springer Berlin / Heidelberg, pp 25–71
4. Brzustowski J Clustering Calculator, `http://www2.biology.ualberta.ca/jbrzusto/cluster.php`
5. Durbin R, Eddy S, Krogh A, Mitchison G (1998) Biological sequence analysis: probabilistic models of proteins and nucleic acids. Cambridge University Press
6. Halkidi M, Batistakis Y, Vazirgiannis M (2001) On clustering validation techniques. Intelligent Information Systems 17(2-3):107–145
7. Hansen P, Jaumard B (1997) Cluster analysis and mathematical programming. Math Program 79(1-3):191–215
8. Henz SR, Huson DH, Auch AF, Nieselt-Struwe K, Schuster SC (2005) Whole-genome prokaryotic phylogeny. Bioinformatics 21(10):2329–2335
9. Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. ACM Comput Surv 31(3):264–323
10. Jones T (1995) Evolutionary algorithms, fitness landscapes and search. PhD thesis, University of New Mexico
11. Jones T, Forrest S (1995) Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithms. In: 6th International Conference on Genetic Algorithms, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp 184–192
12. Kamvar SD, Klein D, Manning CD (2002) Interpreting and Extending Classical Agglomerative Clustering Algorithms using a Model-Based approach. In: 19th International Conference on Machine Learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp 283–290
13. Koljonen J (2006) On fitness distance distributions and correlations, GA performance, and population size of fitness functions with translated optima. In: Honkela T, Kortela J, Raiko T, Valpola H (eds) 9th Scandinavian Conference on Artificial Intelligence, Finnish Artificial Intelligence Society, Espoo, Finland, pp 68–74
14. Kotsiantis S, Pintelas P (2004) Recent Advances in Clustering: A Brief Survey. Transactions on Information Science and Applications 1(1):73–81
15. Krasnogor N, Pelta DA (2004) Measuring the similarity of protein structures by means of the universal similarity metric. Bioinformatics 20(7):1015–1021
16. Krasnogor N, Gustafson S, Pelta D, Verdegay J (2008) Systems Self-Assembly: Multidisciplinary Snapshots, Studies in Multidisciplinarity, vol 5. Elsevier
17. Li L, Siepmann P, Smaldon J, Terrazas G, Krasnogor N (2008) Automated Self-Assembling Programming. In: Krasnogor N, Gustafson S, Pelta D, Verdegay JL (eds) Systems Self-Assembly: Multidisciplinary Snapshots, Elsevier
18. Michielsen K, Raedt HD (2000) Morphological image analysis. Computer Physics Communications 1:94–103
19. Michielsen K, Raedt HD (2001) Integral-geometry morphological image analysis. Physics Reports 347:461–538
20. Quick RJ, Rayward-Smith VJ, Smith GD (1998) Fitness Distance Correlation and Ridge Functions. In: 5th International Conference on Parallel Problem Solving from Nature, Springer-Verlag, London, UK, pp 77–86

21. Rothemund PWK, Winfree E (2000) The program-size complexity of self-assembled squares (extended abstract). In: 32nd ACM symposium on Theory of computing, ACM, New York, NY, USA, pp 459–468
22. Terrazas G, Krasnogor N, Kendall G, Gheorghe M (2005) Automated Tile Design for Self-Assembly Conformations. In: IEEE Congress on Evolutionary Computation, IEEE Press, vol 2, pp 1808–1814
23. Terrazas G, Gheorghe M, Kendall G, Krasnogor N (2007a) Evolving Tiles for Automated Self-Assembly Design. In: IEEE Congress on Evolutionary Computation, IEEE Press, pp 2001–2008
24. Terrazas G, Siepman P, Kendal G, Krasnogor N (2007b) An Evolutionary Methodology for the Automated Design of Cellular Automaton–Based Complex Systems. Journal of Cellular Automata 2(1):77–102
25. Tomassini M, Vanneschi L, Collard P, Clergue M (2005) A Study of Fitness Distance Correlation as a Difficulty Measure in Genetic Programming. Evolutionary Computation 13(2):213–239
26. Vanneschi L, Tomassini M (2003) Pros and Cons of Fitness Distance Correlation in Genetic Programming. In: Barry AM (ed) Bird of a Feather Workshops, Genetic and Evolutionary Computation Conference, AAAI, Chigaco, pp 284–287
27. Vanneschi L, Tomassini M, Collard P, Clergue M (2003) Fitness Distance Correlation in Structural Mutation Genetic Programming. In: Ryan C, Soule T, Keijzer M, Tsang E, Poli R, Costa E (eds) Genetic Programming, Proceedings of EuroGP, Springer-Verlag, Essex, Lecture Notes in Computer Science, vol 2610, pp 455–464
28. Winfree E, Yang X, Seeman NC (1996) Universal computation via self-assembly of DNA: Some theory and experiments. In: DNA Based Computers II, American Mathematical Society, vol 44, pp 191–213