

Towards an Evolved Lower Bound for the Most Circular Partition of a Square

Claudia Obermaier
Artificial Intelligence Research Group
University of Koblenz-Landau
56016 Koblenz, Germany
Email: obermaie@uni-koblenz.de

Markus Wagner
Artificial Intelligence Research Group
University of Koblenz-Landau
56016 Koblenz, Germany
Email: wagnermar@uni-koblenz.de

Abstract—We examine the problem of partitioning a square into convex polygons which are as circular as possible. Circular means that the polygon’s aspect ratio is supposed to be near 1. The aspect ratio of a convex polygon denotes the ratio of the diameters of the smallest circumscribing circle to the largest inscribed disk. This problem has been solved for the equilateral triangle as well as for regular k -gon with $k > 4$. In the case of a square, the optimal solution is still an open problem. We are planning to find a solution which is ”good enough” with the help of evolutionary algorithms.

I. INTRODUCTION

The *aspect ratio* γ of a polygon is the ratio of the diameters of the smallest circumscribing circle to the largest inscribed circle. Since the aspect ratio is a measurement for the circularity, a polygon with a small aspect ratio is also called *fat* ([Dam04],[vdSO94]). When speaking of the aspect ratio of a partition, we mean the maximal aspect ratio of the partition’s pieces. One of the open problems presented at the 14th Canadian Conference on Computational Geometry was to find a partition of a given polygon, which is as circular as possible, i.e., a partition with a minimal aspect ratio. We are interested in a certain restriction of this problem: we consider regular polygons and partitions consisting of convex polygons allowing Steiner points, i.e., the pieces of the partition are allowed to have vertices which are not vertices of the original polygon. For regular k -gons with $k > 4$ it is known ([DIO03]) that the one-piece partition already has the smallest aspect ratio. Further the equilateral triangle has been closely investigated and it has been shown that the most circular partition consists of infinitely many pieces, but it can be approached up to any precision by a finite partition. However the square is still an open problem. It is known that the one-piece partition is not optimal for the square. In [DIO03] it is shown that the most circular partition of the square has an aspect ratio in the range [1.28868, 1.29950]. Figure 1 shows the best solution found so far with $\gamma = 1.29950$, and the value 1.28868 marks the mathematically derived lower bound.

Partitionings of polygons into pieces have been closely investigated in research ([Kei00], [Ber97]). Allowing Steiner points increases the complexity of this problem. In [CD85]

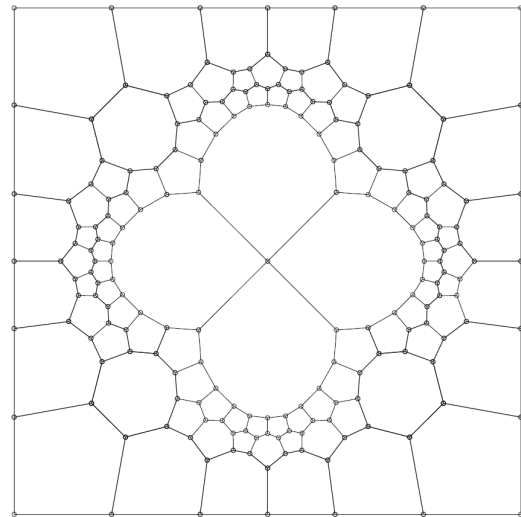


Fig. 1. Best solution found so far in [DIO03], $\gamma = 1.29950$.

a polynomial-time algorithm is introduced for partitioning a polygon into the fewest number of convex pieces.

Collision detection tests are important in motion planning, dynamics simulations as well as virtual reality. It is a known fact ([vdSO94]) that collision detection tests perform notably good on circular polygons since these polygons can be tightly circumscribed by a circle. Further applications for which circular polygons are desirable are ray tracing ([FDFH96]), range searching ([OvdS94]) and simulation of physically-based motion ([HKM96]).

A. Notation

A *partition* of a polygon P is a division into non-overlapping polygonal pieces P_1, P_2, \dots that cover all of P . We will use $\gamma_1(P)$ for the aspect ratio of the one-piece: the ratio of the radius of the smallest circumcircle of P , to the radius of the largest disk inscribed in P . $\gamma(P)$ is the maximum of all the $\gamma_1(P_i)$ for all pieces P_i in a partition of P .

B. Outline

In the following we present an evolutionary algorithm to compute a circular partition of a square. To the best

of our knowledge this problem has not been approached using evolutionary algorithms before. In the next section we give information on the representation of individuals, the fitness function and the selection scheme adopted. Further we introduce three operators we used. Section 3 presents accomplished experiments with results and in the last section we discuss future work.

II. EVOLUTIONARY ALGORITHMS TO COMPUTE A CIRCULAR PARTITION OF A SQUARE

The application of an evolutionary algorithm to a numerical optimization problem is a very common approach. Evolving high-level structures, such as communication networks or time-tables, have been subject to recent studies as well ([MS06], [Fil92]). In order to find a partition of a polygon, an efficient floating-point optimization on the lower level has to be combined with a goal-driven partitioning mechanism on the higher level.

Our research was motivated by open problems that have been solved only partially by conventional analytical methods. For example, the problem of finding longest snakes in hypercubes was posed by [Kau58] and is solved for very few dimensions only. While in the lower dimensions conventional analytical methods were sufficient, the application of evolutionary algorithms to this problem has become successful in finding new lower bounds for higher dimensions ([CP05], [DGH06]).

For the problem at hand, we propose an evolutionary algorithm based on ideas taken from the field of genetic algorithms ([Hol75]) and evolution strategies ([Rec73]). The parameter settings were loosely based upon suggestions by [Gol89].

A. Individual Representation, Fitness Function and Selection

1) *Individual Representation*: Finding an adequate representation proved to be a crucial step in the development of the entire evolutionary algorithm. A structure had to be found that fulfilled several requirements:

- It must be capable of representing unambiguous valid partitions,
- it should allow the construction of complex evolutionary operators,
- it should support arbitrary numbers of vertices, edges and polygons per partition.

Based on the evaluation of the scenario, we chose an object-oriented approach, which is drafted in Figure 2. In the figure, a vector denotes a list-like data-structure that contains arbitrary objects. An individual (i.e. a partition) is composed of an arbitrary number of polygons. A single polygon is composed of an arbitrary number of edges, which are defined by two vertices. Furthermore, a vertex has references to all the edges it belongs to. The same applies to the edges, which have information about the polygons they belong to. While most edges belong to two polygons, an edge that belongs to the very border of the outer square belongs to only

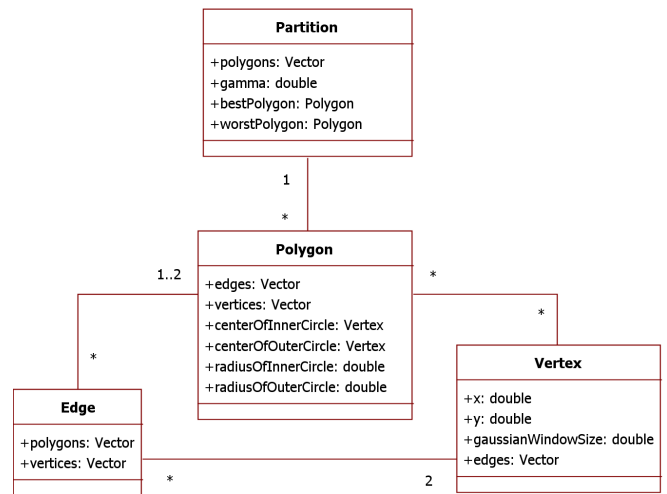


Fig. 2. Schematic of an individual's general structure.

one polygon. This overall top-down method allows for easy navigation in the partition, for example for finding adjacent polygons of a given vertex.

2) *Fitness Function*: The problem can be regarded to be multi-objective, since the optimal solution is likely to be symmetric. Thus, not only the polygons have to be as circular as possible, but also symmetry is desirable. The following two fitness functions were alternated every 20 generations:

$$f_1(P) = \text{maximum}(\gamma_1(P_i)) \quad (1)$$

$$f_2(P) = \text{mean}(\gamma_1(P_i)) \quad (2)$$

P is the partition and the P_i s are the pieces of the partition. f_1 evaluates the quality of a partition by concentrating solely on the least circular polygon. This produces a higher pressure to improve the worst polygon of the current partition. f_2 left some room to optimize other polygons and to introduce symmetry. Polygons that were more circular than the worst polygon could be optimized as well, creating “room” for the following 20 generations when f_1 would be used again.

3) *Selection Mechanisms*: The individuals for the application of the variation operators (see Section II.B) were selected using tournament selection, with a tournament size of two individuals and an uncertainty of 90%. This means that the fitter individual out of two randomly drawn ones was chosen with a probability of 90%, leaving a probability of 10% to preserve diversity within the population and possibly prevent early convergence on a local optimum.

Based on the initial μ individuals, a number of $\lambda = 7 * \mu$ individuals per generation was created and their fitnesses calculated. Then, the best μ individuals were chosen deterministically from the union of parents and offspring.

B. Variation Operators

1) *Push Operator*: In order to modify the coordinates of a single vertex at a time, two different approaches were taken. For the first approach, the Gaussian perturbation by

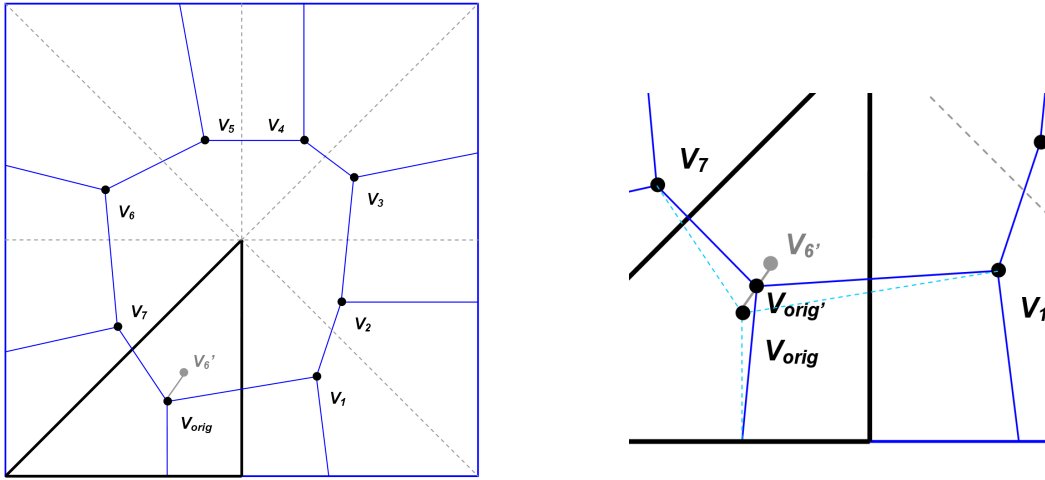


Fig. 3. Example of an application of the push operator, using other vertices as a heuristic. The right figure shows details of the mutated partition.

[BFM00] was adapted. This mutation operator for floating-point numbers is based on a Gaussian distribution, whose mean is the current value of a coordinate and the standard deviation σ . In a process of self-adaption, the value of σ is adapted every k generations, based on the 1/5 success rule [Rec73]. This allows for mutation to use a wider search space when mutation is successful. In the other case, the standard deviation is decreased to concentrate the search more around the current solution.

The second approach for adjusting the position of a vertex implicitly introduced symmetry by taking *similar* vertices into account. The idea is that the basic square can be divided into eight triangles. The triangles cover parts of the partition that can be transformed into one another by reflections¹—taking an acceptable degree of precision into account. The application is demonstrated in Figure 3. V_{orig} is the vertex to be mutated and belongs to the least circular polygon of the individual. Vertices $V_1 \dots V_7$ are considered similar to V_{orig} , and V_6 is randomly chosen to serve as a heuristic. Subsequently, V_{orig}' is set on a point between V_{orig} and the transformed vertex V_6' . In the case that V_{orig} would always be set to the transformed vertex, the partition would quickly be highly symmetrical. However, it is likely that it would lose good partial solutions for the entire partition of the square.

2) *Tile Operator*: The second operator adds vertices to a polygon P . First we choose a polygon P' which is adjacent to P and whose number of vertices n is not a prime number. Let p be the smallest prime factor of n . In the next step we mark the mutual edge of P and P' and starting from this edge we mark every p -th edge of P' . Let M be the center of the largest inscribed circle of P' and E_i be the center of the i -th marked edge in P' . We now construct subsidiary lines from each E_i to M . In Figure 4 on the left side, those lines

are depicted as dashed lines within P' . In the next step a new vertex V_i on each subsidiary line is constructed.

We now change all polygons whose mutual edge to P' has been marked by adding the respective vertex V_i to those polygons. Since P is one of those polygons, P gets a new vertex. Further we remove polygon P' and add $\frac{n}{p}$ new polygons. The right part of 4 shows the result. So the tile operator adds one vertex to $\frac{n}{p}$ polygons, removes one polygon and creates $\frac{n}{p}$ new polygons each having $n + (p - 2)$ vertices.

Since the newly created polygons are supposed to be as circular as possible, it is important to start the tile operator by marking edges in P' , which are uniformly distributed. This can be best achieved by choosing p to be a divider of n . This explains, why n is not allowed to be a prime number. In addition to the uniform distribution of marked edges it is also desirable only to have a small number of consecutive edges which are not marked. Having many consecutive edges which are not marked is likely to result in the creation of longish polygons with a high aspect ratio. Therefore we decided to use the smallest prime factor of n as p .

3) *Star Operator*: During the evolution process it is possible that a concave piece occurs in the partition. If many other pieces in this partition have a very small aspect ratio, it is not desirable to discard the whole individual. Therefore we developed an operator which allows to repair such a partition. However this operator can not handle all kinds of concave polygons. It can only be used to get rid of benign concave polygons. We call a concave polygon with vertices P_1, P_2, \dots, P_n *benign concave*, if no segment from P_i to the center of the largest inscribed circle subtends one of the polygons edges.

Let P be a benign concave polygon with vertices P_1, P_2, \dots, P_n occurring in a partition and M be the center of its largest inscribed circle. Further with E_i we denote the center of the i -th edge. In the first step, subsidiary lines from each E_i to M and from each P_i to M are constructed. Next, new vertices are created: on each subsidiary line connecting P_i

¹Theoretically, this idea could have been exploited in the representation design, giving us the phenotype's symmetry "for free". The downside is that the application of the variation operators to elements on/near the triangles borders would have been algorithmically extremely complicated.

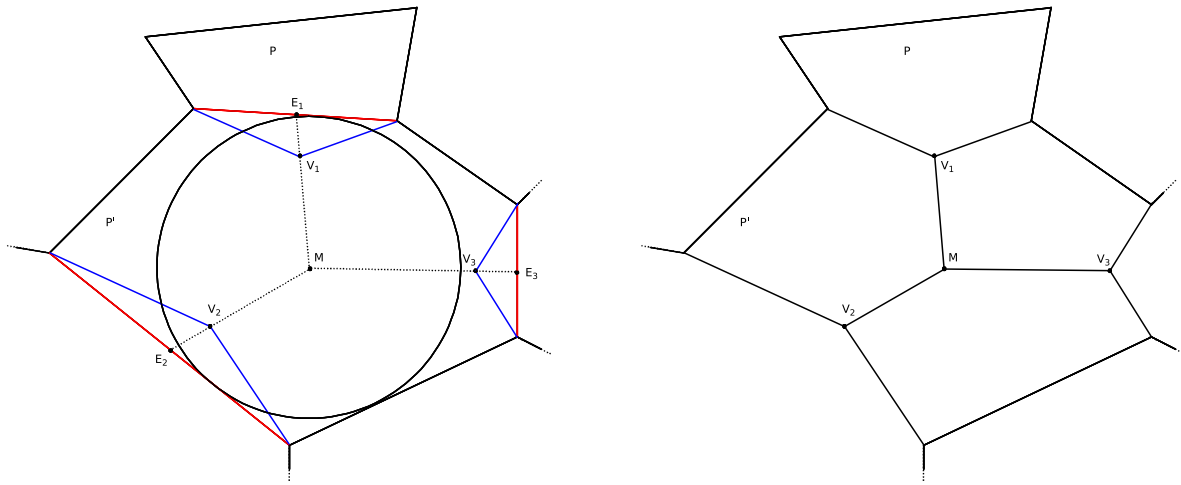


Fig. 4. First steps of an application of the tile operator on the left, result on the right.

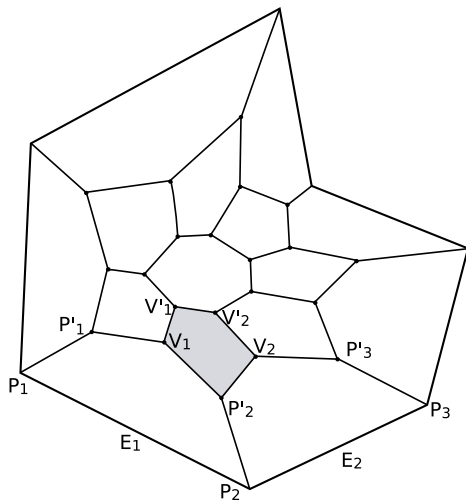


Fig. 6. Result of the star operator.

and M a new vertex P'_i and on each subsidiary line from E_i to M a new vertex V_i is created. The exact position of a vertex P'_i complies with the length of P 's edges containing P_i . Further the position of V_i is determined by the positions of P'_i and P'_{i+1} . The left side of Figure 5 depicts P , the new vertices as well as the subsidiary lines.

In the next step the new vertices are connected, creating a concave star-shaped polygon in the middle of the original polygon. Further we create n new pentagons around the star-shaped polygon. The result is depicted on the right side of Figure 5, where one of the new pentagons is shaded.

In the next step, on each segment $V_i M$ a new vertex V'_i is created. Those new vertices are connected creating an n -gon in the middle. Further we also use the new vertices to create n new pentagons, one in each pinnacles of the star shaped polygon. Figure ?? depicts the result.

4) *Crossover Operator*: The definition of an applicable crossover operator was not possible. An analysis of the general approach lead to the problem of inevitably leaving the valid search space. Even in the case of the combination of two good individuals, additional vertices might have to be introduced, which would lead to concave polygons and invalid solutions. In general, the matching and mating of graphs is another difficult problem—even without taking a degree of impression into account. We tried to compensate the lack of an explicit crossover operator by the introduction of the push operator that has already been defined in this section.

C. Initial Population

To improve upon the one-piece partition of the square, polygons with at least five vertices must be used, because the square itself is the most circular tetragon with $\gamma = 1.4142$. Covering the sides of the square is challenging and crucial to the structure of the overall partition. Therefore, we chose to use different classes of seeds that were constructed manually, each class having different characteristics regarding the sides of the square. Figure 7 shows two examples that both have near-optimal polygons on the sides. Their central polygons are concave and have to be partitioned into convex polygons by the variation operators in order to follow the problem given.

III. EXPERIMENTAL RESULTS

We split up the experiments into sets of short runs and long runs. In the first set, the effect of the introduced operators was studied. For the long runs, the experiments were not a priori limited in run-time in order to eventually come up with a new lower bound for the problem.

Considering the parameter settings, the push operator was used for 1 to 5 vertices at a time, with a rate of 0.2. There was a 50:50 chance of either mutating an arbitrary vertex or one of the vertices that belonged to the least circular polygon. The initial standard deviation of a vertex was set

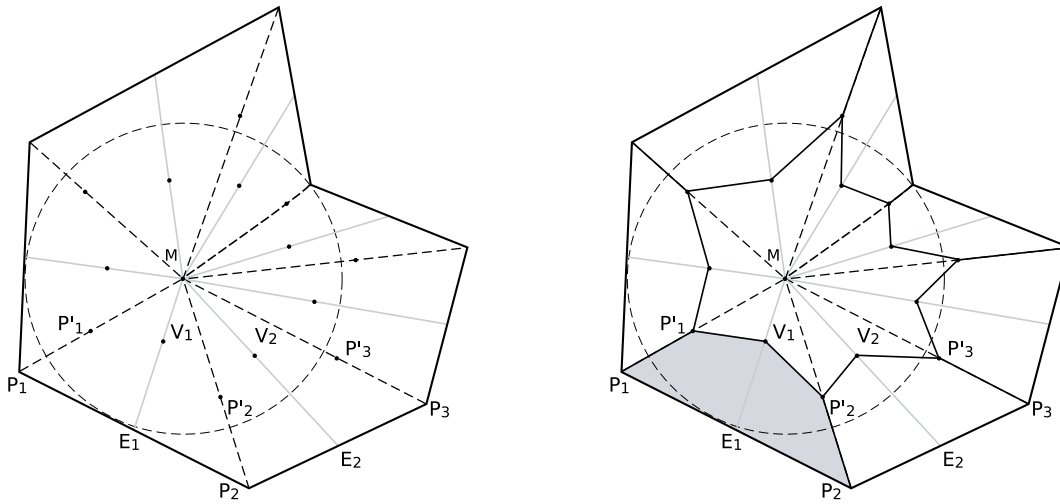


Fig. 5. First step of an application of the star operator on the left side, second step on the right side.

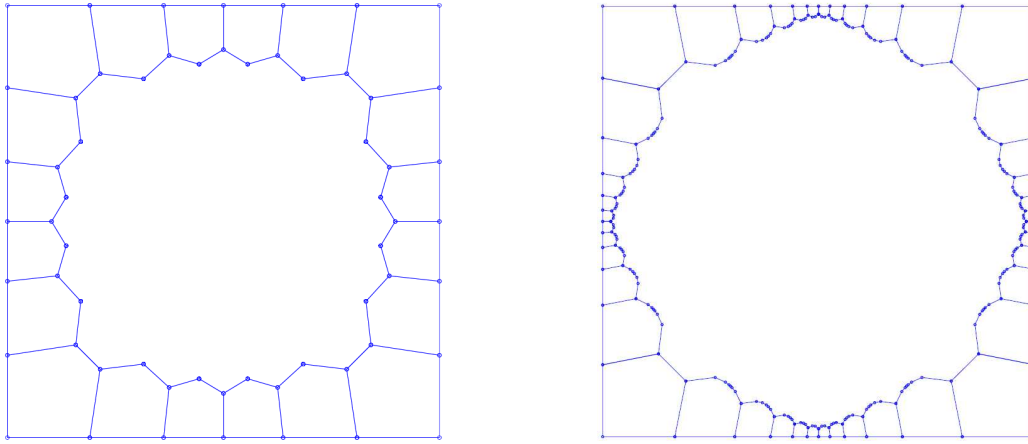


Fig. 7. Examples for complex seeds used in the initial population of the long run experiments.

to $\sigma = 0.02$. The tile and star operators were checked for applicability at rates of 0.1 each.

A. Short run experiments

For the evaluation of the mutation operators, the evolutionary algorithm was tested with three different setups:

- EA_1 : basic setup, with the mutation of a single vertex being limited to a predefined interval,
- EA_2 : same as EA_1 , with the Gaussian perturbation being used by the mutation of a single vertex,
- EA_3 : same as EA_2 , using the implicit introduction of symmetry.

Figure 8 shows an example for an individual that was used as a seed. The other seeds follow that general structure, having pentagons in the corners of the square, tetragons in between, and an octagon in the center.

Each setup was repeated 30 times, with a population size $\mu = 10$. Evolution was stopped after 2000 generations, which

is equivalent to 140,000 evaluations. Figure 10 shows the average fitnesses—averaged over the individual runs—for the setups EA_1 to EA_3 .

The best individual found during our first series of tests is shown in Figure 9. It's fitness of $\gamma = 1.3405$ was found by EA_3 that used both the Gaussian perturbation and the operator for the implicit introduction of symmetry. Regarding the average performance as well as the best solutions, the setup EA_3 outperformed the setups EA_1 and EA_2 , whose best individuals found had fitnesses of $\gamma_{EA_1} = 1.3419$ resp. $\gamma_{EA_2} = 1.3427$. To our knowledge, the best partition with a structure similar to Figure 9, has a fitness of $\gamma = 1.3396$ is optimal.

B. Long run experiments

For the long runs, we used several approaches to seed the population. We used seeds that follow the structure of Figure 8, as well as seeds similar to Figure 1 and intermediate partitions. Figure 1 has a fitness of $\gamma = 1.2995$, which is the established lower bound by [DIO03]. It features a total of 185

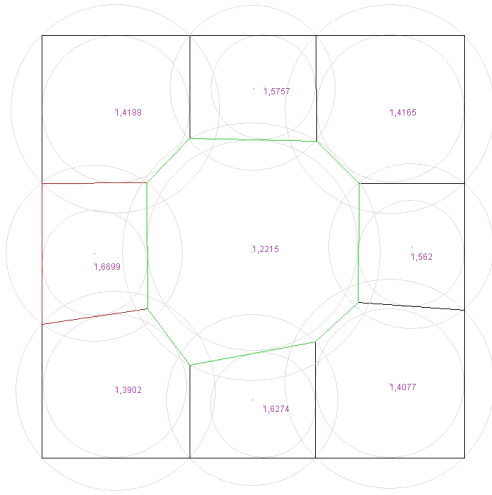


Fig. 8. Example of an initial partition of the short run experiments, $\gamma = 1.6699$.

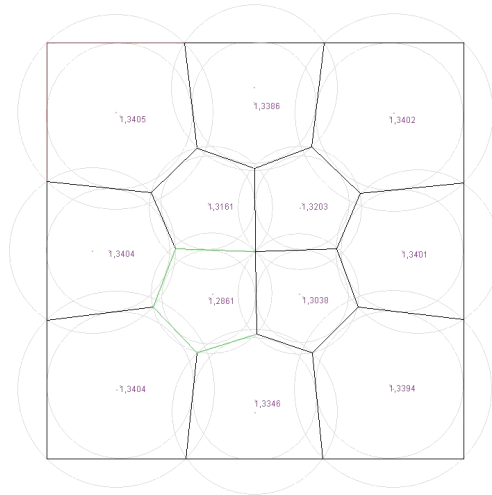


Fig. 9. Best solution found by the short run experiments, $\gamma = 1.3405$.

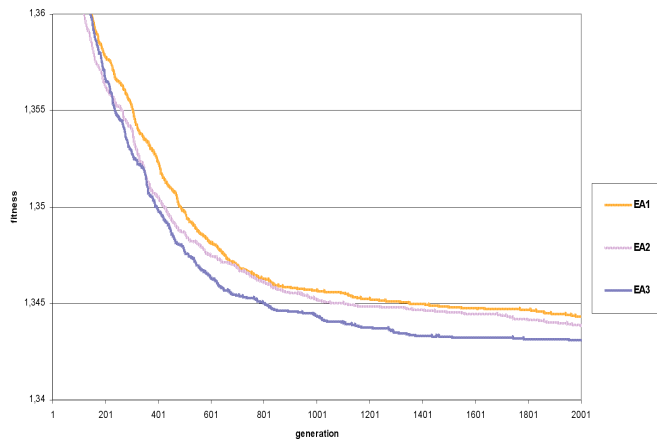


Fig. 10. Comparison of different setups for the push operator.

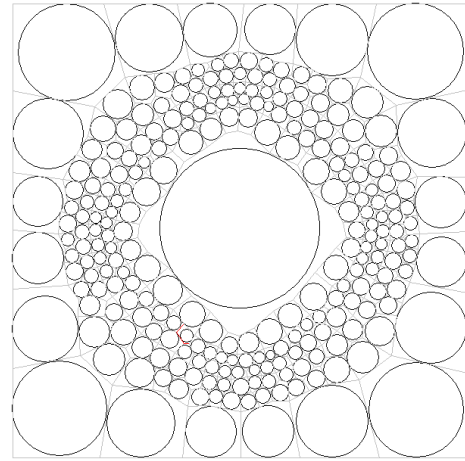


Fig. 11. Complex individual, $\gamma = 1.425$. The largest inscribed disks were highlighted to demonstrate that the problem is not the coverage of the square with circles, but the partitioning into pieces with a specific property.

vertices, 96 edges and 92 polygons. Starting with this seed left the evolutionary algorithm with $185 * 2 = 370$ floating-point parameters per individual to optimize. Furthermore, the structure of the partition is represented by the edge- and polygon-relations over the vertices, making an individual an immensely complex combination of a large number of floating-point parameters and relations between these.

Figure 11 shows an example of an individual where the star operator has been applied to several times. It is important to note that the higher value for γ is not the result of the large uncovered areas around the central circle. The problem lies in the tight packing of circular polygons in the rings. No improvement over the results of the short runs have been made, using the more complex seeds.

C. Discussion

Even though we used sophisticated operators, our experiments did not result in a partition with a γ value lower than the one of the best partition introduced in [DIO03]. There are several reasons for this: The seeding we used forces the individuals to develop in a certain way and therefore huge parts of the search space remain unexplored. The individuals get stuck in areas of local minima and it turns out that it is not trivial to leave them. The long run experiments imply that using more complex seeds does not solve this problem. One possibility to leave a local minima would be an operator, which could be used to insert points by splitting edges. However this leads almost inevitably to concave polygons, and thus to invalid partitions which will be discarded by selection immediately. Therefore, the algorithms operators pervasively comply with the constraint to always form only convex partitions.

Further, as already mentioned, an individual is an immensely complex combination of a large number of floating-point parameters and relations between these. These relations cause the entire system to be highly epistatic, resulting

in a very low ability to construct higher-order building blocks from lower-order building blocks. In [Fog95] it is hypothesized that recombination performs poorly in systems that are extensively pleiotropic and highly polygenic, and it is argued that mutation would be superior. Our observations coincide with this hypothesis. Even though we rely heavily on the effects of the sophisticated mutation operators, the satisfying results of the short run experiments support our approach.

The definition of the problem itself, that is minimizing the maximum of the aspect ratios of a square's partition, showed to include a conceptual problem. Since the fitness value of an individual solely depends on the worst aspect ratio, there is almost no possibility to take the rest of the individual into consideration as well. During the experiments, we observed that in early generations, the algorithm usually concentrated on the improvement of a specific polygon before switching over to another polygon. In later generations, the algorithm's optimization oscillated between several polygons, yielding little improvements regarding the fitness of the individual.

IV. CONCLUSION AND FUTURE WORK

In this paper we addressed the problem of partitioning a square into circular and convex polygons using an evolutionary algorithm. We identified the theoretical problems and solved them by introducing specialized operators. The applicability of the features was justified and the increase in performance was confirmed by experiments. However, no new lower bound was found so far.

Observations made during the experiments showed that finding an optimal aspect ratio partition of a square is not straightforward. In order to lower the established lower bound, it would be necessary to use convex pieces with seven or more vertices on the square's sides. Figures 9 and 11 indicate that the use of such pieces creates large discrepancies in the edge lengths of the interior polygons. Although filling the interior becomes problematic, it is not supposed to be fundamentally impossible. Nevertheless, the system proved to be very difficult to optimize because it is highly epistatic.

We will continue our research and concentrate on improving our operators by using predefined patterns. These templates will work as local heuristics, i.e., suggestions for the partitioning of polygons of similar shape.

REFERENCES

- [Ber97] Marshall W. Bern. Triangulations. In J.E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, pages 413–428. CRC Press LLC, 1997.
- [BFM00] Thomas Bäck, David B. Fogel, and Zbigniew Michalewicz, editors. *Evolutionary Computation 2: Advanced Algorithms and Operators*. Institute of Physics Publishing, Bristol, 2000. contains excerpts from the Handbook of Evolutionary Computation.
- [CD85] Bernard Chazelle and David P. Dobkin. Optimal convex decompositions. In G. T. Toussaint, editor, *Computational Geometry*, pages 63–133. North-Holland, 1985.
- [CP05] Darren A. Casella and Walter D. Potter. New lower bounds for the snake-in-the-box problem: Using evolutionary techniques to hunt for snakes. In Ingrid Russell and Zdravko Markov, editors, *FLAIRS Conference*, pages 264–269. AAAI Press, 2005.
- [Dam04] Mirela Damian. Exact and approximation algorithms for computing optimal fat decompositions. *Comput. Geom.*, 28(1):19–27, 2004.
- [DGH06] Pedro A. Diaz-Gomez and Dean F. Hougen. Genetic algorithms for hunting snakes in hypercubes: Fitness function analysis and open questions. In Yeong-Tae Song, Chao Lu, and Roger Lee, editors, *SNPD*, pages 389–394. IEEE Computer Society, 2006.
- [DIO03] Mirela Damian-Iordache and Joseph O'Rourke. Partitioning regular polygons into circular pieces I: Convex partitions. *CoRR*, cs.CG/0304023, 2003. informal publication.
- [Fog95] David B. Fogel. *Evolutionary Computation: toward a new philosophy of machine intelligence*. IEEE Press, 1995.
- [FDFH96] James D. Foley, Andries v. Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics: Principles and Practices (2nd edition in C)*. Addison Wesley, 1996.
- [Fil92] Bogdan Filipic. Enhancing genetic search to schedule a production unit. In *ECAI*, pages 603–607, 1992.
- [Gol89] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.
- [HKM96] Martin Held, James T. Klosowski, and Joseph S. B. Mitchell. Real-time collision detection for motion simulation within complex environments. In *ACM Siggraph Vis. Proc.*, page 151. ACM, 1996.
- [Hol75] John H. Holland. *Adaption in Natural and Artificial Systems*. University of Michigan Press, 1975.
- [Kau58] W. H. Kautz. Unit-distance error-checking codes. In *IRE Transactions*, pages 179–180, 1958.
- [Kei00] J. Mark Keil. Polygon decomposition. In J.-R. Sack and J. Urratia, editors, *Handbook of Computational Geometry*, pages 491–518. Elsevier, 2000.
- [MS06] C. Mala and S. Selvakumar. Construction of an optimal multicast tree for group communication in a cellular network using genetic algorithm. *Computer Communications*, 29(16):3306–3312, 2006.
- [OvdS94] Mark H. Overmars and A. Frank van der Stappen. Range searching and point location among fat objects. In *European Symposium on Algorithms*, pages 240–253, 1994.
- [Rec73] Ingo Rechenberg. *Evolutionsstrategie*, volume 15 of *problemata*. Friedrich Frommann Verlag (Günther Holzboog KG), Stuttgart, 1973.
- [vdSO94] A. Frank van der Stappen and Mark H. Overmars. Motion planning amidst fat obstacles (extended abstract). In *Symposium on Computational Geometry*, pages 31–40, 1994.