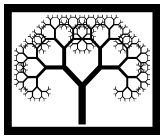
Paper 220



Proceedings of the Eleventh International Conference
on Computational Structures Technology,
B.H.V. Topping, (Editor),
Civil-Comp Press, Stirlingshire, Scotland

Shape Optimization using Isogeometric Analysis and Particle Swarm Optimization

A. Pospíšilová, M. Lepš, D. Rypl and B. Patzák
Department of Mechanics, Faculty of Civil Engineering
Czech Technical University in Prague, Czech Republic

Abstract

This paper describes the application of the particle swarm optimization to the shape optimization of two-dimensional domains described by NURBS (non-uniform rational B-splines) and analyzed using the NURBS-based isogeometric analysis. The regularization of the optimization problem, preventing undesirable clustering of control points of the underlying geometry leading to invalid geometry or parametrization, is achieved by controlling the magnitude of perturbation of design variables within the PSO using a background mesh. This mesh, however, does not have to comply with requirements on a standard (e.g. finite element) computational mesh, as it does not have to follow the exact geometry. Thus construction of such a mesh (Matlab `Distmesh` tool is utilized) is simple and does not introduce a bottleneck to the whole process. The capabilities and performance of the developed optimization strategy are demonstrated on standard benchmark problems.

Keywords: shape optimization, particle swarm optimization, NURBS, isogeometric analysis, `Distmesh` tool.

1 Introduction

Isogeometric analysis (IGA) [1, 2, 3, 4, 5, 6] is a recently introduced method which builds upon the concept of isoparametric elements and upgrades it to the geometry level. Although the original intention was to span the gap between the computer aided design (CAD) and the finite element method (FEM), the various advantages and range of applicability make the IGA an interesting alternative to the widely used FEM. It has been shown that the IGA outperforms the classical FEM in various aspects (accuracy, robustness, system condition number, etc.). Another distinct advantage of the IGA over the FEM consists in the conciseness of the parametrization of the design variable

space, which makes the IGA attractive for the shape optimization problems.

The aim of this paper is to present the application of the particle swarm optimization (PSO) to the shape optimization of two-dimensional domains described by NURBS and analyzed by the NURBS-based IGA. The regularization of the optimization problem, preventing undesirable clustering of control points of the underlying geometry leading to invalid geometry or parametrization, is achieved by controlling the magnitude of perturbation of design variables within PSO using a background mesh.

The paper is organized as follows. The concept of the IGA is briefly recalled in Section 2. A method of the Particle Swarm Optimization is introduced in Section 3 and the `Distmesh` tool for mesh generating is shortly mentioned in Section 4. The final combination of all these methods is described in Section 5 and results are presented in Section 6.

2 Isogeometric Analysis

In the IGA, the approximation of the solution over the domain is based on the functions employed for the description of the underlying geometry of the domain itself. Therefore understanding of the NURBS based representation of the geometry (used in CAD) gives a good insight into the isogeometric concept. A NURBS patch is defined by a set of control points (topologically forming a regular grid of the dimension corresponding to the spatial dimension of the underlying parametric space), their weights, degree of the B-spline basis functions in each direction of the parametric space, and a so-called knot vector represented by a nondecreasing sequence of parametric coordinates for each direction defining the support for individual B-spline basis functions (in other words parametrization) in that particular direction. Note that the number of control points, degree of basis functions, and size of the knot vector in the particular parametric direction are not independent and must be mutually consistent. The data at the control points (for example the coordinates when the geometry is concerned, or the primary unknowns when the solution space is handled) are interpolated over the NURBS patch using the shape functions which are defined as weighted normalized tensor product of univariate B-spline basis functions in each of the parametric directions. For example, for a two-dimensional NURBS patch of a degree p in u-direction and degree q in v-direction, the basis function associated with a control point in the i-th row and j-th column of the grid of $N \times M$ control points is given by

$$R_{i,j}^{p,q}(u,v) = \frac{N_i^p(u) N_j^q(v)\, w_{i,j}}{\sum_{n=1}^{N} \sum_{m=1}^{M} N_n^p(u) N_m^q(v)\, w_{n,m}}, \tag{1}$$

where $w_{i,j}$ stands for the control point weight and $N_k^r(t)$ denotes the univariate B-spline basis functions of the degree r. Starting with the piecewise constant basis functions of a zero degree defined by

$$N_i^0(t) = \begin{cases} 1 & \text{if } t_i \le t < t_{i+1}, \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

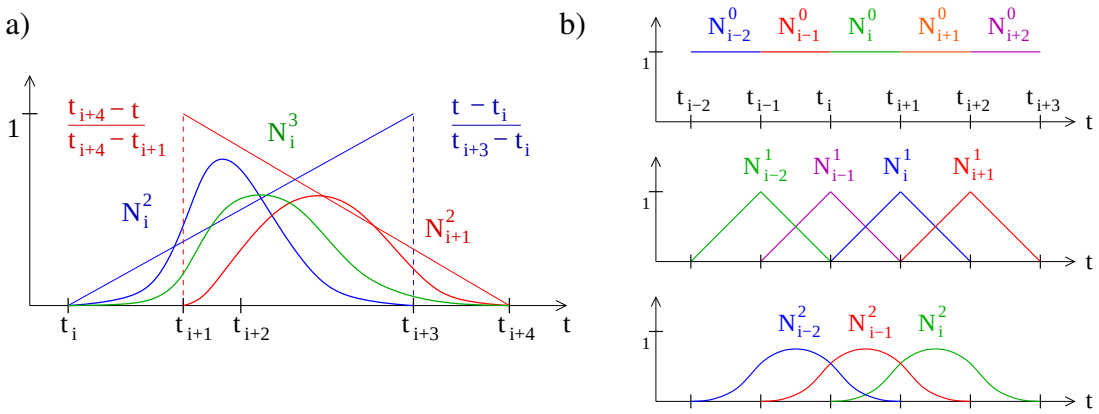


Figure 1: B-spline basis functions: (a) construction of cubic basis function as linear combination of quadratic basis functions, (b) hierarchical sequence of piecewise constant, linear, and quadratic basis functions.

the basis functions for degree $p > 0$ are defined recursively as

$$N_i^p(t) = \frac{t - t_i}{t_{i+p} - t_i} N_i^{p-1}(t) + \frac{t_{i+p+1} - t}{t_{i+p+1} - t_{i+1}} N_{i+1}^{p-1}(t), \tag{3}$$

in which t_i (for $i = 1, 2, \ldots, N + p + 1$) stands for entries of the knot vector and N denotes the number of control points (in the given direction). This is demonstrated in Figure 1a) where the cubic basis function N_i^3 spanning four consecutive knot spans is obtained as linear combination of consecutive quadratic basis functions N_i^2 and N_{i+1}^2 spanning the first three and last three from those four knot spans, respectively. Figure 1b) then displays the hierarchical sequence for piecewise constant, linear, and quadratic basis functions built over an infinite uniform knot vector. For details concerning the definition of the B-spline basis functions and their properties the reader is referred to [7].

An example of a quadratic NURBS curve (i.e. one-dimensional NURBS patch) defined by six control points and their weights and parameterized over the open knot vector[1] $\{0, 0, 0, 1, 3, 3, 4, 4, 4\}$ is depicted in Figure 2a). The parametric equation of that particular curve is given by

$$\boldsymbol{r}(t) = \sum_{i=1}^{6} R_i(t) \boldsymbol{P}_i, \tag{4}$$

where $\boldsymbol{r}$ is the positional vector of a point on the curve corresponding to parameters $t \in \langle 0, 4 \rangle$ and $\boldsymbol{P}_i$ represents the individual control points. The colors of individual parts of the curve correspond to the individual non-zero knot spans (red: $0 - 1$, green: $1 - 3$, blue: $3 - 4$). The NURBS basis functions $R_i^2(t)$ as well as the B-spline basis

[1]Knot vector is called open if its first and last entry is repeated (degree + 1) times, which implies that the curve is passing through the first and last control point (see [7] for details).

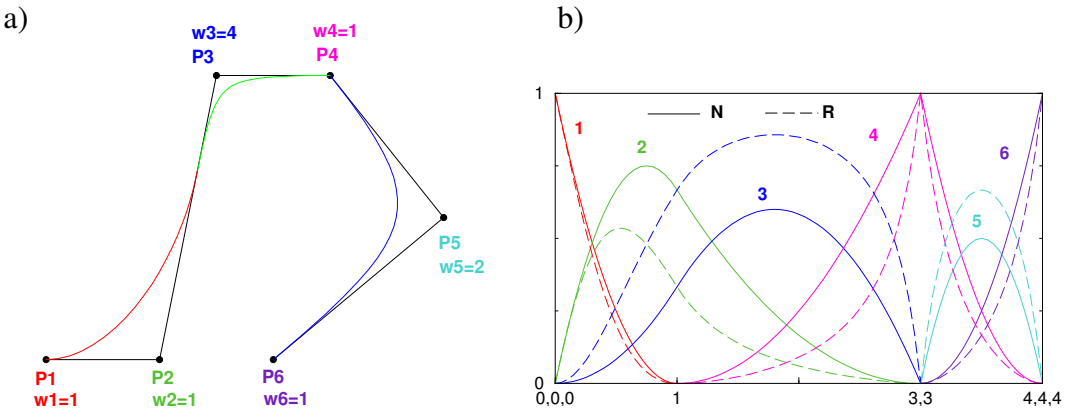


Figure 2: Quadratic NURBS curve: (a) control polygon in black; numbers of individual control points and their weights (in parenthesis) in color corresponding to associated basis function; segments of the curve in red/green/blue corresponding to non-zero knot spans 0-1/1-3/3-4, (b) B-spline basis functions N_i and NURBS basis functions R_i corresponding to individual control points plotted over the entire span of the knot vector $\{0, 0, 0, 1, 3, 3, 4, 4, 4\}$.

functions $N_i^2(t)$ used to construct $R_i^2(t)$ are shown in Figure 2b) over the entire span of the knot vector. The curve interpolates those control points for which the corresponding basis function attains value one (knot value at which this occurs defines the parameter corresponding to that control point), the rest of the control polygon is only approximated. The curve is C^1 continuous everywhere except for the point corresponding to parameter 3 at which the continuity has been weakened by repeating that particular value in the knot vector twice.[2] Note the C^0 continuity of the B-spline basis function N_4 in Figure 2b) at parameter 3. The coincidence of the interface between the first (red) and the second (green) knot span on the curve with the intersection of the curve with its control polygon is a rule for quadratic curve only. Note that the red part of the curve (corresponding to the first knot span of size 1) is significantly larger than the green part (corresponding to the second knot span of size 2) despite the fact that the control polygon between control points 1 and 4 is symmetric with respect to the middle of its second segment. This is the consequence of the weight 4 applied at the third control point which results in the attraction of the curve toward the third control point.

The computational isogeometric mesh within the single NURBS patch is formed by partitioning the parametric space into the non-zero knot spans in each direction (in the example above, there are three such non-zero knot spans, see Figure 2). Since the shape functions within the single non-zero knot span are C^∞, the computation of characteristic components of the discretized governing differential equation (e.g. stiffness matrix, load vector, etc.) on each non-zero knot span is performed in the

[2]Generally, multiplicity $k \leq p$ of a particular inner knot decreases the continuity of the basis functions of degree p at that knot to C^{p-k}.

standard FE-like fashion, typically using the Gaussian numerical quadrature[3].

The IGA has many features in common with the FEM (the shape functions form a partition of unity, they have the compact support, affine invariance applies, a numerical integration is employed, Neumann boundary conditions are satisfied naturally etc.) but there are some more or less significant differences. In the traditional FEM, the individual nodes are part of the computational domain, and corresponding degrees of freedom (DOFs) have the direct physical meaning (e.g. displacement in particular direction at the node), which is the direct consequence of the Kronecker delta property of the finite element shape functions. In the framework of the IGA, the control points of NURBS patches are generally not part of the physical computational domain. This implies that the application of Dirichlet boundary conditions is not straightforward and must be handled (often only approximately) within the available NURBS space. Except for the h-, p-, and hp-refinement strategies, the isogeometric concept offers also a higher order refinement methodology, known as k-refinement [1, 4], which has no analogue in the standard FEM and which is based on the fact that knot insertion (refinement of the parametric space) and degree elevation algorithms do not commute. Using the k-refinement, it is possible to increase the continuity across knot span boundaries (within a single NURBS patch) while limiting the growth of control variables. An important feature of the IGA analysis is the fact that due to a larger support of basis functions of quadratic degree and higher[4] the number of control points necessary to obtain results of similar quality as that from the FEM using a basis function of the same degree, is smaller. The same also holds for the representation of the underlying geometry. This conciseness of parametrization makes the IGA attractive for the shape optimization problems as the size of the design variable space is kept limited while still preserving sufficient level of flexibility and geometrical continuity. Moreover, the approximation property of the B-spline basis functions (the actual geometry does generally not interpolate individual control points) eliminates the undesirable oscillations (know in the FEM) due to the interpolation nature of (typically Lagrangian) finite element basis functions and reduces the need for the regularization.

3 Particle Swarm Optimization

A lot of new metaheuristic optimization methods is inspired by nature. Algorithms simulate social behavior of animals, birds or insects or behave according to some physical phenomena. One of the main advantages of metaheuristics is that there is no need to determine a gradient of an objective function as in case of mathematical programming methods. Particle Swarm Optimization, as a member of Swarm Intelligence techniques, is a relatively new method firstly introduced in [8]. It is based on a natural behavior of bird flocking or fish schooling. The flock acts like one organism. The whole flock as well as every individual, called particle, has its own memory.

[3]Note, however, that Gaussian numerical quadrature is generally not optimal and that there exist more efficient numerical integration schemes for the IGA.

[4]Note, that for linear degree, the IGA analysis is identical with the FEM based on linear elements.

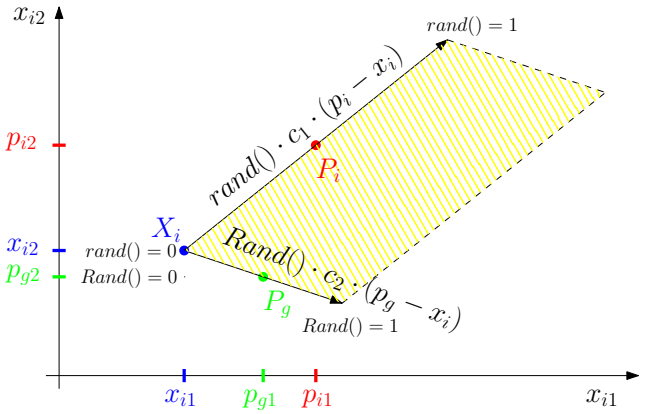


Figure 3: Velocity update of one particle in 2D where $c_1 = c_2 = 2$.

Particles share their best knowledge to each other thus the flock can find the optimal solution efficiently.

Each particle represents a potential solution in D-dimensional space. The i-th particle X_i is represented as $X_i = (x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iD})$. Particles fly through the searched space with a velocity $V_i = (v_{i1}, v_{i2}, v_{i3}, \ldots, v_{iD})$. Every component of the velocity is updated according to the following equation

$$v_{id}^{j+1} = w \cdot v_{id}^{j} + c_1 \cdot rand() \cdot (p_{id} - x_{id}^{j}) + c_2 \cdot Rand() \cdot (p_{gd} - x_{id}^{j}), \quad d = i, \ldots, D \quad (5)$$

where w is the inertia weight, v_{id}^{j} is a velocity from the previous step, c_1 is the cognitive factor, c_2 is the social factor, p_{id} represents the best position of the particle i, p_{gd} represents the best swarm position and $rand()$ and $Rand()$ are two random scalars, vectors or matrices, respectively, in the range of [0,1] (see [9]). The first addend of the Equation 5 is deterministic and represents an inertia of the particle. If there is a zero contribution from the second as well as the third addend, the particle will move only with inertia from the previous step and it does not stay in a local minimum. The second addend of Equation 5 is stochastic because of the random function $rand()$. It provides movement of the particle towards its own best position. The third contribution is stochastic as well and provides shifting of the particle to the best swarm position. Figure 3 illustrates stochastic contributions of Equation 5. Since each addend is scaled by a random number in the range of [0,1], the end of the velocity vector can be placed anywhere in the yellow hatched area.

The position of the particle x_{id}^{j} is then updated by

$$x_{id}^{j+1} = x_{id}^{j} + 1 \cdot v_{id}^{j+1} \quad (6)$$

where v_{id}^{j+1} is an actual velocity from Equation 5 and 1 has the meaning of the unit time.

There are two possibilities how to scale stochastic addends [10]. The first approach, a linear PSO, is using two random scalars as $rand()$ and $Rand()$ which multiply the

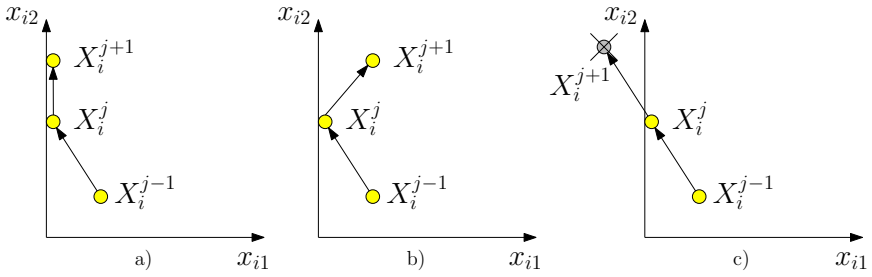


Figure 4: Types of walls - (a) absorbing wall, (b) reflecting wall, (c) invisible wall.

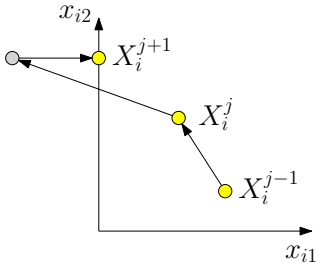


Figure 5: Returning process for a particle which flies out of the solution space.

magnitude of the cognitive and social vectors. The main disadvantage of this method is that particle will fly over the straight line at the end of the algorithm and whole space will not be searched. The second approach, a classical PSO, produces a two random vectors or two random diagonal matrices, respectively. The movement of particles is then diverse during the whole run. The classical approach is used hereafter.

The algorithm of the PSO can be described as follows.

1. The first step is to set up all coefficients and variables. We use values as listed in Table 1, see the next section for the discussion on the settings. Note that initial velocities can be zero or random. Non-zero velocities have a merit of diversity in the deterministic addend, and therefore, have been used in our implementation.

2. In the next step, the value of an objective function is calculated for all particles. The value is then compared with the best solution of the particle and with the best solution of the whole swarm. In case of better value in the actual iteration than in previous ones, the positions P_{gd} and P_{id} are updated.

3. Velocities are calculated for all particles according to Equation 5 and positions are updated by Equation 6.

4. The algorithm ends after reaching a maximum number of iterations selected at the first step.

The main drawback is that the PSO can move particles out of the admissible space. The easiest solution is to restrict a velocity or coordinates for a particle which flies out-

side. The three most frequent PSO approaches of controlling velocitiesare depicted in Figure 4. An *absorbing wall* ensures the movement of the particle along the boundary, a *reflecting wall* bounces the particle back to the admissible space and an *invisible wall* does not utilize updating of a particle position if the movement directs behind the boundary. In our implementation, another approach is used. The coordinates are limited to the given bounds, i.e. if the particle flies out of the admissible space, it is returned back to the boundary according to Figure 5.

4 Mesh generation using $Distmesh$ tool

The `Distmesh` tool (DM) is a heuristic smoothing algorithm for generating uniform meshes [11]. The DM is based on a simple dynamical system of expanding pin-jointed structure, here characterized by the second (background) mesh, see Figure 6a). Those trusses that are too short are causing repulsive forces that move the too close nodes apart, see Figure 6b) for the final solution. The main disadvantage apart from high computational demands is the need to return nodes that leave the prescribed admissible domain, i.e. the same problem encountered within the PSO. The DM offers similar procedure as is shown in Figure 5 for basic entities. A polygon used in our computations to describe the background mesh boundary is one of them, see the original paper [12] for more details. The background mesh, however, does not have to comply with requirements on a standard (e.g. FEM) computational mesh, as it does not have to precisely follow the exact geometry. Thus construction of such mesh can be simplified up to only two inner iterations of the DM and thus does not introduce a bottleneck to the whole process.

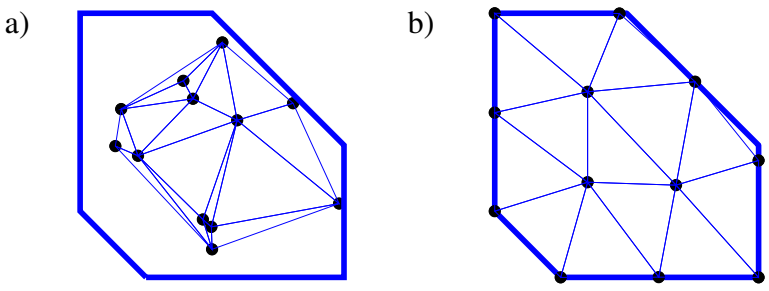


Figure 6: An illustrative example of a generation of a uniform mesh from randomly generated points inside a polygon: (a) Triangulation of random points forming a truss-like structure, (b) The final mesh after the application of the `Distmesh` tool.

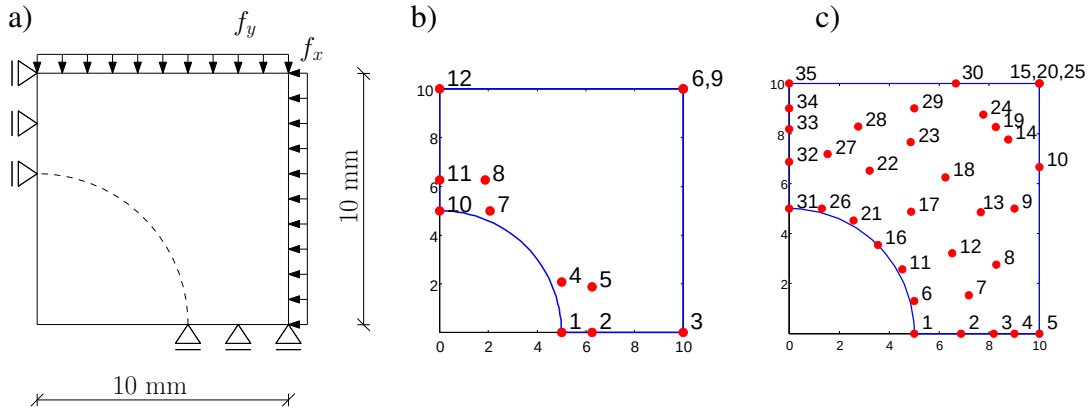


Figure 7: A square plate with an indicated hole (a) and starting positions of control nodes for (b) coarse and (c) fine settings.

5 Shape optimization method

A combination of all methods mentioned above is applied on the benchmark structure depicted in Figure 7a) taken from [13]. The symmetrically supported plate is 10 mm high as well as wide and the thickness is 1 mm. Young's modulus E is equal to 10 N/mm^2 and Poisson ratio ν equals to 0.3. Control nodes, placed on the structure, define a curve of a hole and a boundary polygon. The results will be shown for two resolutions of the control points, coarse and fine settings, see Figure 7b) and Figure 7c).

The overall objective of the shape optimization is to find positions of the control nodes so that the shape of the benchmark structure has minimal compliance in the discretized form [14]

$$\begin{aligned} & \min \ \mathcal{L} && = \tfrac{1}{2} f^T u, && (7)\\ s.t. \quad & \boldsymbol{K} u && = f, && (8)\\ s.t. \quad & V && = const., && (9) \end{aligned}$$

with the stiffness matrix $\boldsymbol{K}$ and u and f as displacements and load vectors, respectively. Two examples of loadings and volume constraints are solved. The first example is symmetrically stretched, $f_x = f_y = -1$ N/mm and the total volume is restricted to 70% of the original volume, i.e. to $V = 70$ mm^2. The second example is characterized by an unsymmetrical loading $f_x = -1$ N/mm, $f_y = -2$ N/mm and a volume constraint is set to $V = 95$ mm^2. Note that OOFEM software ([15]) enhanced by the Isogeometric analysis [16] is used for the computation of the compliance $\mathcal{L}$ and the volume V of the primary mesh as well.

The single objective optimization problem is therefore composed of the objective function $\mathcal{L}$ enhanced by the penalty function ensuring fulfilment of the volume con-

straint. The general shape of the penalty function used hereafter reads as

$$penalty_i = \left(\frac{\chi_i}{\alpha_{min}}\right)^{\beta} * MaxPen_i, \tag{10}$$

where χ_i is the value of the i-th equality constraint, violation of which by α_{min} percent is penalized with the $MaxPen_i$ value; a parameter β then influences the steepness of the penalty curve. Particularly, in case of the volume restriction, the penalty function is defined by

$$\chi_1 = \frac{V}{V_{max}} - 1, \quad MaxPen_1 = 2, \tag{11}$$

where V is the volume obtained from OOFEM using NURBS and V_{max} is the requested volume mentioned above. Other parameters are listed in Table 1.

PSO					Penalty functions	
c_1	c_2	w	$iter$	nop	α_{min}	β
2	1	0.6	100	16	0.005	2

Table 1: Coefficients for the PSO are following: c_1 is a cognitive factor, c_2 is a social factor, w is an inertia weight, $iter$ is the number of iterations, nop is the number of particles. Penalty functions have following coefficients: α_{min} is a distance where $MaxPen_i$ penalty is assigned and β is a shape parameter.

Then, one particle X_i in the PSO represents one potential solution. Since the plate is in 2D and has N control points, the vector X_i contains its all coordinates i.e. $2N$ components. The number of particles can be set to relatively low value e.g. 16 and the number of iterations can be restricted to 100. A cognitive factor is set to $c_1 = 2$ according to [10] and a social factor to $c_2 = 1$. The second factor is set to lower value than in [10] because we use a less number of particles and the social knowledge is not that important as the personal knowledge, i.e. we would like to keep the diversity of solutions. For the sake of completeness, all important coefficients are listed in Table 1.

Some nodes defining the boundary are forbidden to move because of prescribed supports. It is therefore necessary to distinguish which components of velocities can be zeros and non-zeros, respectively. If a movement of the node is allowed, the corresponding component in the velocity vector is at the start of the PSO set to a random value in the range of $[-1, 1]$. Otherwise, the component is zero. The initial positions of nodes are sums of starting positions depicted in Figure 7b) and Figure 7c) and initial velocities according to Equation 6.

In the next step of the algorithm the `Distmesh` tool is used. It ensures that the nodes do not move towards each other by limiting the maximal length of the velocity terms to the half of the shortest edge connecting the given node within the background mesh. However, this does not ensure transposing of nodes as shown in Figure 8a)

and forming of unwanted loops on the boundary. This problem is solved by another penalty term

$$\chi_2 = V - V_{DM} \quad MaxPen_2 = 2, \tag{12}$$

where V is the volume obtained from Isogeometric analysis and V_{DM} is the volume obtained from the DM background mesh; i.e. we penalize big differences between these two meshes in terms of volumes, see Figure 8c), which is usually the case of the loops on the boundary.

The last deficiency is an appearance of a peak with almost zero volume in pursuit of spreading out the points, see Figure 8b). This is again solved by the penalty approach in terms of perpendicularity of quoins emerged in points 1 and 10 in Figure 7b). Penalty function is used in the following form

$$\chi_3 = \sum_{j=1}^{2} \cos\phi_j, \quad MaxPen_3 = 0.1, \tag{13}$$

where ϕ_j is an angle of the boundary at points 1 and 10, respectively.

6 Results

The known analytical optima of the selected benchmark are characterized by the ellipsoidal hole with different radii a and b for the prescribed loadings and a volume. Note that in case of $f_x = f_y$ the hole is circular. Therefore, we have simulated the problem with two parameters a and b in the range of [1,9] with step 0.1. The situation is depicted in Figure 9a). The coordinates of curve's control nodes are proportionally changed thus this simple model does not cover all possible solutions. The compliance for all values of parameters is depicted in Figure 9b) and Figure 9c) for the first and second example, respectively. The white lines show solutions with a proper prescribed volumes. The objective function is minimized thus the optimum is the solution with

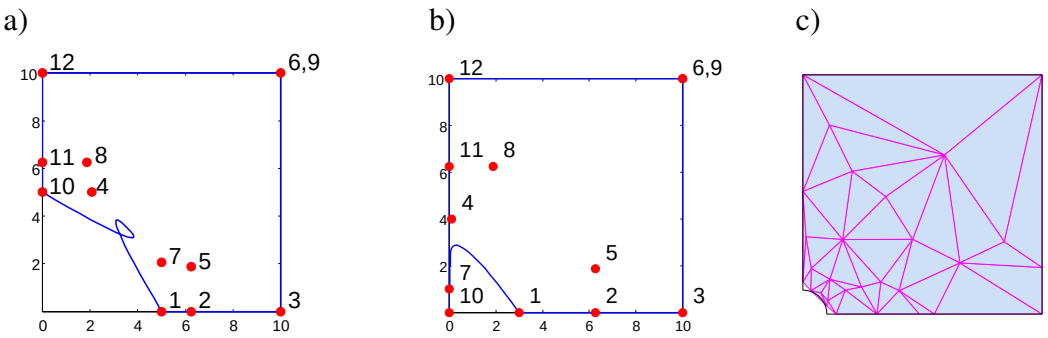


Figure 8: (a) A loop which is created in case two nodes are transposed, (b) irregularity in corner nodes and (c) a difference between a real NURBS mesh and a background mesh from the `Distmesh` tool.

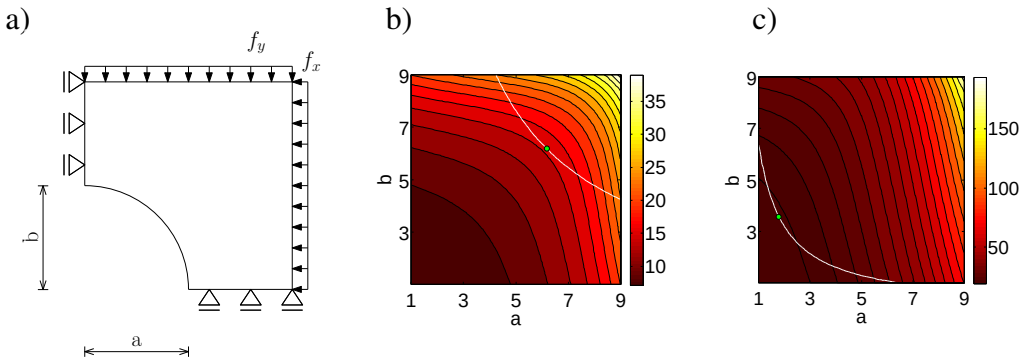


Figure 9: (a) A parametric solution of benchmarks: compliance of parametric solutions where a and b are in the range of [1,9] with a step of 0.1 mm and white lines represent prescribed volumes V with the green points of analytical optima for the first example (b) $V = 70$ mm^2 and a diameter of 6.18 mm and the second example (c) $V = 95$ mm^2 and $a = 1.8$ mm and $b = 3.6$ mm.

the minimal compliance value lying on the curve given by those white lines, here represented with the green points.

author	method	compliance	volume
[13]	mesh 16x16	16.165 Nmm	-
[13]	mesh 32x32	16.193 Nmm	-
[13]	mesh 64x64	16.195 Nmm	-
this paper	parametric solution - coarse settings	15.529 Nmm	69.809 mm^2
this paper	presented method - coarse settings	15.574 Nmm	69.702 mm^2
this paper	parametric solution - fine settings	16.156 Nmm	69.809 mm^2
this paper	presented method - fine settings	16.047 Nmm	70.013 mm^2

Table 2: Comparison of results for reference [13] and the presented method for the first example.

Four optima of the benchmark presented in Figure 10 are obtained with the proposed method described in Section 5. The obtained values of the objective functions and constraints are in reasonable agreement with theoretical results, see comparison of values in Table 2 and Table 3 for the optima obtained by the parametric solution based on an analytical solution and by the presented approach. However, the obtained graphical solutions can deviate from theoretical results for fine resolutions, see again Figure 10. With increasing refinement of the problem, some pathological be-

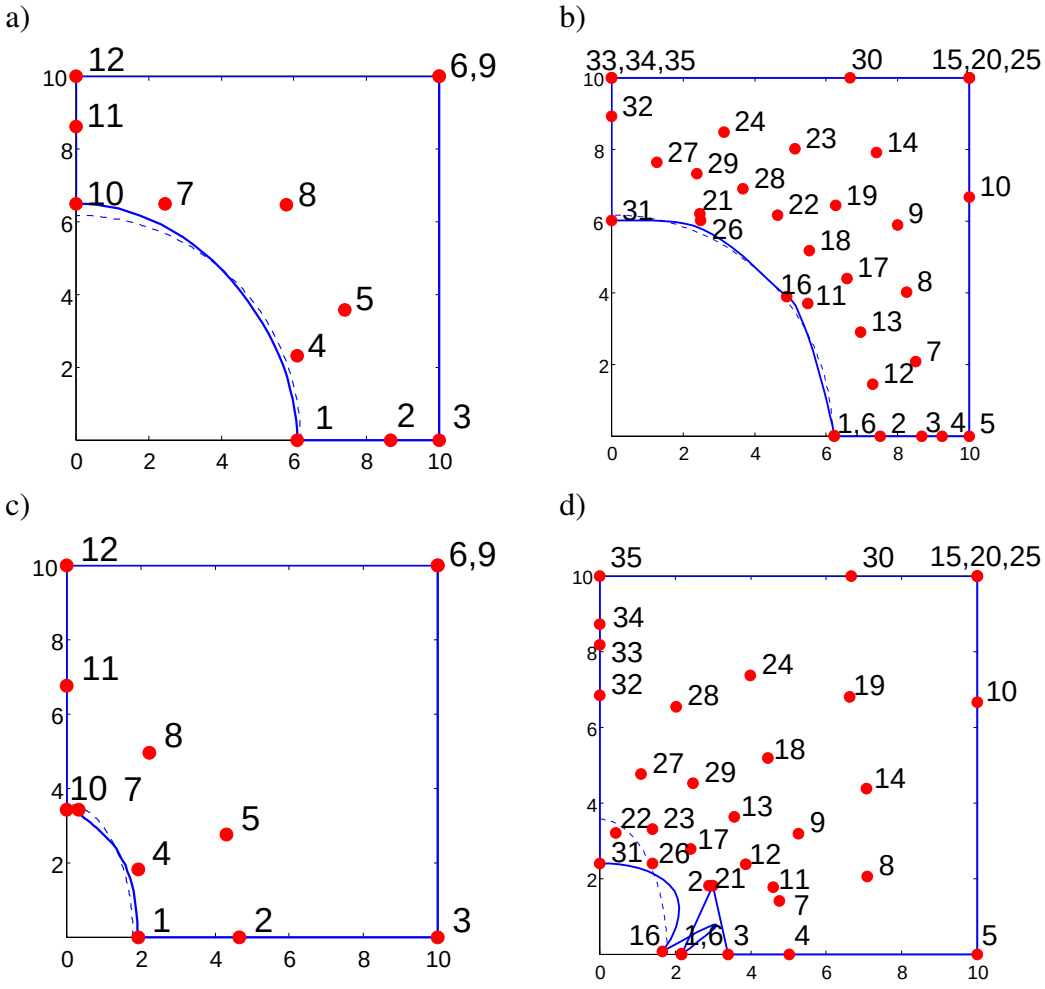


Figure 10: Best solutions obtained by the PSO for the coarse (Left column) and fine settings (Right column) and for the first (Top) and the second (Bottom) example. The dashed lines represent shapes of analytical optima.

haviour is observed. Possible causes may be not large enough size of the considered population, enforced boundary conditions (to keep the hole smooth) or the rather not optimal behaviour of `Distmesh` along boundary. All these aspects need to be investigated in the future research.

7 Conclusion

This paper has shown a combination of three methods. The Isogeometric analysis is a step towards a CAD method which, as an addendum, has several advantages over the classical FE analysis in obtaining the mechanical response of a structure. The precise description of the geometry predetermines IGA as a solution to the shape optimiza-

author	method	compliance	volume
[13]	mesh 16x16	$21.337\ \mathrm{Nmm}$	-
[13]	mesh 32x32	$21.377\ \mathrm{Nmm}$	-
[13]	mesh 64x64	$21.389\ \mathrm{Nmm}$	-
this paper	parametric solution - coarse settings	$21.331\ \mathrm{Nmm}$	$94.911\ \mathrm{mm}^2$
this paper	presented method - coarse settings	$21.338\ \mathrm{Nmm}$	$94.991\ \mathrm{mm}^2$
this paper	parametric solution - fine settings	$21.402\ \mathrm{Nmm}$	$94.911\ \mathrm{mm}^2$
this paper	presented method - fine settings	$21.091\ \mathrm{Nmm}$	$95.255\ \mathrm{mm}^2$

Table 3: Comparison of results for reference [13] and the presented method for the second example.

tion problem. The particle wwarm optimization algorithm is then characterized by a physical meaning of a group of flying particles which can utilize the inner properties of the dynamics of the particles. The shape optimization problem is difficult from the regularity point of view. Therefore, not only limitations within the PSO have been used in this paper, but also a second, background mesh produced by the `Distmesh` tool has been utilized. The solutions obtained indicate the problem of irregularities with a growing number of control points arises from the unsatisfactory description of the optimization solutions near the boundaries.

Acknowledgments

This work was supported by the Grant Agency of the Czech Republic - Projects No. 103/09/2009 and No. P105/12/1146. Their financial assistance is gratefully acknowledged.

References

[1] T.J.R. Hughes, J.A. Cottrell, Y. Bazilevs, “Isogeometric Analysis: CAD, Finite Elements, NURBS, Exact Geometry and Mesh Refinement”, Computer Methods in Applied Mechanics and Engineering, 194, 4135–4195, 2005.

[2] Y. Bazilevs, L. Beirao de Veiga, J.A. Cottrell, G. Sangalli, “Isogeometric Analysis: Approximation, Stability and Error Estimates for h-refined Meshes”, Mathematical Models and Methods in Applied Sciences, 16 (7), 1031–1090, 2006.

[3] J.A. Cottrell, A. Reali, Y. Bazilevs, T.J.R. Hughes, “Isogeometric Analysis of Structural Vibrations”, Computer Methods in Applied Mechanics and Engineering, 195, 5257–5296, 2006.
[4] J.A. Cottrell, T.J.R. Hughes, A. Reali, “Studies of Refinement and Continuity in Isogeometric Structural Analysis”, Computer Methods in Applied Mechanics and Engineering, 196, 4160–4183, 2007.
[5] J.A. Cottrell, T.J.R. Hughes, Y. Bazilevs, Isogeometric Analysis: Toward Integration of CAD and FEA, John Wiley & Sons, 2009.
[6] Y. Zhang, Y. Bazilevs, S. Goswami, C. Bajaj, T.J.R. Hughes, “Patient-specific Vascular NURBS Modeling for Isogeometric Analysis of Blood Flow”, Computer Methods in Applied Mechanics and Engineering, 196, 2943–2959, 2007.
[7] L. Piegl, W. Tiller, The NURBS Book, Springer-Verlag, 1997.
[8] J. Kennedy, R.C. Eberhart, R.C., “Particle Swarm Optimization”, in “IEEE International Conference of Neural Networks”, 4, 1942–1948, 1995.
[9] Y. Shi, R. Eberhart, “A Modified Particle Swarm Optimizer”, in “IEEE International Conference on Evolutionary Computation”, 69-73, 1998.
[10] D.N. Wilke, S. Kok, A.A. Groenwold, “Comparison of Linear and Classical Velocity Update Rules in Particle Swarm Optimization: Notes on Diversity”, International Journal for Numerical Methods in Engineering, 70 (8), 962–984, 2007.
[11] L. Chen, M. Holst, “Efficient Mesh Optimization Schemes Based on Optimal Delaunay Triangulations” Computer Methods in Applied Mechanics and Engineering, 200 (9–12), 967–984, 2011.
[12] P.-O. Persson, G. Strang, “A Simple Mesh Generator in MATLAB”, SIAM Review, 46 (2), 329–345, 2004.
[13] J. Norato, R. Haber, D. Tortorelli, M. P. Bendsøe, “A Geometry Projection Method for Shape Optimization”, International Journal for Numerical Methods in Engineering, 60 (14), 2289–2312, 2004.
[14] M. P. Bendsøe, O. Sigmund, Topology Optimization: Theory, Methods and Applications. Springer-Verlag.
[15] B. Patzák, “OOFEM project home page”, http://www.oofem.org, 2012.
[16] D. Rypl, B. Patzák, “From the Finite Element Analysis to the Isogeometric Analysis in an Object Oriented Computing Environment”, Advances in Engineering Software, 44 (1), 116–125, 2012.