Journal of Global Optimization **16:** 109–131, 2000.

Genetic and Random Search Methods in Optimal Shape Design Problems

J. HASLINGER[1], D. JEDELSKÝ[2], T. KOZUBEK[3] and J. TVRDÍK[4]
[1]*KFK MFF UK, Ke Karlovu 5, 121 16 Praha 2, CZ*
(E-mail: haslin@apollo.karlov.mff.cuni.cz); [2]*Institute for Research and Applications of Fuzzy Modelling, University of Ostrava, Bráfova 7, 701 03 Ostrava 1, CZ;* [3]*Dept. of Computer Science, VŠB-Technical Univ., tř. 17. listopadu, 708 33 Ostrava 8, CZ;* [4]*Dept. of Computer Science, Univ. of Ostrava, Bráfova 7, 701 03 Ostrava 1, CZ*

(Received 27 April 1998; accepted in revised form 29 June 1999)

Abstract. We describe the application of two global optimization methods, namely of genetic and random search type algorithms in shape optimization. When the so-called fictitious domain approaches are used for the numerical realization of state problems, the resulting minimized function is non-differentiable and stair-wise, in general. Such complicated behaviour excludes the use of classical local methods. Specific modifications of the above-mentioned global methods for our class of problems are described. Numerical results of several model examples computed by different variants of genetic and random search type algorithms are discussed.

Key words: Genetic algorithm, Control random search method, Shape optimization, Fictitious domain approaches

1. Introduction

The aim of this paper is to show how genetic and stochastic type global optimization methods may be helpful in the numerical realization of optimal shape design problems. A traditional way how shape optimization problems are realized is based on the so called *boundary variation technique*. This technique, besides of advantages has also some drawbacks. One of the most serious ones is the fact that the inner optimization level (i.e. the numerical realization of state problems) is not too efficient. One of possible cures to increase the efficiency is to use the so called *fictitious domain solvers* (FDS's), by means of which state problems can be solved very quickly. As usually, we have to pay for that. The use of FDS's increases the efficiency of the inner optimization level, but (unfortunately) it gives rise to certain pathologies of minimized functions, excluding the use of local optimization methods. A natural idea arises, namely to combine FDS's (inner level) together with global optimization methods (outer level). The main goal of the paper is to present a class of global optimization methods which can be used with the succes in such a type of optimization problems.

The paper is organized as follows: in Section 2 we recall an abstract formulation of a class of optimal shape design problems and their approximations. Section 3 gives a brief survey of two types of fictitious domain methods, based on the duality and the optimal control approach, respectively. We present their finite element approximations and discuss properties of finite element solutions which play the substantional role in the optimization process. In Section 4 two global optimization methods are mentioned, namely *a genetic algorithm* and *the modified control random search* approach. Their specific modifications, important for the numerical realization of our class of problems are presented. Finally, Section 5 presents and compares the results of several model examples computed by using different versions of above mentioned methods.

2. Abstract Setting of Optimal Shape Design Problems and Their Approximations

In practice we often meet problems, in which the shape of a structure is one of the most decisive factors for the quality of the final product (automobile, aerospace industry, ...). Thus it is not surprising that optimal control problems in which the shape is one of control variables, attracted the interest of many applied mathematicians and engineers. Since shape optimization can be viewed to be a special branch of the optimal control theory, its abstract setting is very similar to other optimization problems. A large class of optimal shape design problems can be stated as follows:

$$(\mathbf{P}) \qquad \begin{cases} \text{Find } \Omega^* \in \mathcal{O} \text{ such that} \\ J\big(\Omega^*, u(\Omega^*)\big) = \min\limits_{\Omega \in \mathcal{O}} J\big(\Omega, u(\Omega)\big), \end{cases}$$

where Ω is a domain, playing the role of the *control variable* belonging to a family $\mathcal{O}$ of *admissible domains* in $\mathbb{R}^m$. Further, $u(\Omega)$ is a solution of a *state problem* $\big(\mathcal{P}(\Omega)\big)$, assumed to be a function of Ω. Finally, J is a *cost functional*, the explicit choice of which depends on optimization targets. Domain Ω^* (if it exists) will be called an *optimal* one. The existence of solutions to $(\mathbf{P})$ is now very well established (see Pironneau 1984; Haslinger & Neittaanmäki 1996).

To be able to solve $(\mathbf{P})$ numerically, one has to pass to its approximation. Firstly, the family $\mathcal{O}$ of admissible domains is replaced by another family, denoted by $\mathcal{O}_h$, whose all elements (domains) are determined by the same, finite number of parameters ($\mathcal{O}_h$ contains domains with spline boundaries, e.g.). Thus any domain $\Omega_h \in \mathcal{O}_h$ can be uniquely described by a vector $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_q) \in \mathbb{R}^q$ called the vector of *discrete design variables*. Using this concept, one can introduce the set $\mathcal{U} \subseteq \mathbb{R}^q$ and the isomorphism $\mathcal{T}_D\colon\ \mathcal{O}_h \leftrightarrow \mathcal{U}$ as follows:

$$\begin{aligned} \mathcal{T}_D(\Omega_h) &= \boldsymbol{\alpha}, \quad \Omega_h \in \mathcal{O}_h, \\ \mathcal{T}_D(\mathcal{O}_h) &= \mathcal{U}, \end{aligned}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^q$ is the vector of the discrete design variables, describing Ω_h. The state problem $\big(\mathcal{P}(\Omega)\big)$ will be approximated by using finite elements, e.g. Its finite element approximation will be denoted by $\big(\mathcal{P}(\Omega_h)\big)_h$ and the corresponding solution by $u_h(\Omega_h)$, in what follows.

The *approximation* $\big(\mathbf{P}\big)_h$ of $\big(\mathbf{P}\big)$ now reads as follows:

$$\big(\mathbf{P}\big)_h \qquad \begin{cases} \text{Find } \Omega_h^* \in \mathcal{O}_h \text{ such that} \\ J\big(\Omega_h^*, u_h(\Omega_h^*)\big) = \min\limits_{\Omega_h \in \mathcal{O}_h} J\big(\Omega_h, u_h(\Omega_h)\big). \end{cases}$$

Under reasonable assumptions one can prove the relation between solutions of $\big(\mathbf{P}\big)$ and $\big(\mathbf{P}\big)_h$, when $h \to 0+$ (see Haslinger & Neittaanmäki 1996).

Standard optimization methods used for solving $\big(\mathbf{P}\big)_h$ are based on the construction of *a minimizing sequence* $\big\{\Omega_h^{(k)}\big\}_{k=1}^{\infty}$, i.e. a sequence satisfying

$$J\big(\Omega_h^{(k+1)}, u_h(\Omega_h^{(k+1)})\big) \leqslant J\big(\Omega_h^{(k)}, u_h(\Omega_h^{(k)})\big), \quad k = 0, 1, \dots . \tag{2.1}$$

To find such a sequence, the boundary variation technique uses the method of successive deformations of the previous shape:

$$\Omega_h^{(k+1)} = \mathcal{F}_h^{(k)}(\Omega_h^{(k)}), \quad k = 0, 1, \dots ,$$

where $\mathcal{F}_h^{(k)} : \mathbb{R}^m \to \mathbb{R}^m$ is an one-to-one continuous mapping chosen in such a way that (2.1) is satisfied.

Now let state problem $\big(\mathcal{P}(\Omega)\big)$ be linear and let the standard finite element approach be used for getting $\big(\mathcal{P}(\Omega_h)\big)_h$. Then the algebraic form of $\big(\mathcal{P}(\Omega_h)\big)_h$ leads to the following linear algebraic system:

$$\mathbb{A}(\boldsymbol{\alpha})\mathbf{u}(\boldsymbol{\alpha}) = \mathbf{F}(\boldsymbol{\alpha}), \tag{2.2}$$

where $\mathbb{A}$ is the *stiffness matrix* and $\mathbf{F}$ is the *load vector*. Both, $\mathbb{A}$ and $\mathbf{F}$ depend on the vector of the discrete design variables. The vector $\mathbf{u}(\boldsymbol{\alpha}) \in \mathbb{R}^n$ being the solution of (2.2) is termed the *nodal value vector*. As usually, one can define the isomorphism $\mathcal{T}_S$ associating with the finite element solution $u_h(\Omega_h)$ its nodal value vector $\mathbf{u}(\boldsymbol{\alpha})$:

$$\mathcal{T}_S\big(u_h(\Omega_h)\big) = \mathbf{u}(\boldsymbol{\alpha})$$

and $\mathbf{u}(\boldsymbol{\alpha}) \in \mathbb{R}^n$ solves (2.2).

The algebraic representation of $\big(\mathbf{P}\big)_h$ now reads as follows ($h > 0$ fixed):

$$\big(\vec{\mathbf{P}}\big) \qquad \begin{cases} \text{Find } \boldsymbol{\alpha}^* \in \mathcal{U} \text{ such that} \\ \mathcal{J}\big(\boldsymbol{\alpha}^*, \mathbf{u}(\boldsymbol{\alpha}^*)\big) = \min\limits_{\boldsymbol{\alpha} \in \mathcal{U}} \mathcal{J}\big(\boldsymbol{\alpha}, \mathbf{u}(\boldsymbol{\alpha})\big), \end{cases}$$

with $\mathbf{u}(\boldsymbol{\alpha})$ being the solution of (2.2) and

$$\mathcal{J}\big(\boldsymbol{\alpha}, \mathbf{u}(\boldsymbol{\alpha})\big) \equiv J\big(\mathcal{T}_D^{-1}\boldsymbol{\alpha}, \mathcal{T}_S^{-1}\mathbf{u}(\boldsymbol{\alpha})\big),$$

where the symbols $\mathcal{T}_D^{-1}$, $\mathcal{T}_S^{-1}$ denote the inverse mappings to $\mathcal{T}_D$, $\mathcal{T}_S$, respectively. $(\vec{\mathbf{P}})$ is a *non-linear mathematical programming problem*, in general. From its form, the main shortcomings of such a formulation are readily seen: in order to solve (2.2), one has to update the stiffness matrix $\mathbb{A}$ and the load vector $\mathbf{F}$ after *any* change of α. This, among others, requires to construct a new partition of any new configuration. The whole procedure is repeated many times, after any change of the geometry of Ω_h.

In order to increase the efficiency of the inner level, we propose to use FDS's as a tool for the numerical realization of $\left(\mathcal{P}(\Omega_h)\right)_h$. As we shall see in the next section, fictitious domain approaches enable us to perform all computations on a *fixed* domain $\hat{\Omega}$ and on a *fixed* partition $\hat{\mathcal{R}}_h$ of $\hat{\Omega}$, which are completely independent on the geometry of Ω. As a result, the stiffness matrix $\mathbb{A}$ *will not depend* on the vector of discrete design variables.

3. Fictitious Domain Methods and Their Use in Shape Optimization

The idea of any fictitious domain approach is the same: let us have an elliptic problem

$$(\mathcal{P}) \qquad \begin{cases} Au = f & \text{in } \Omega \\ +b.c. & \text{on } \partial\Omega, \end{cases}$$

where $\Omega \subset \mathbb{R}^2$ is a domain, possibly with a complicated shape. We plug Ω into a new domain $\hat{\Omega}$ (termed the *fictitious* one), having a simple shape (a rectangle, e.g.). Now we define a new problem $(\hat{\mathcal{P}})$ in $\hat{\Omega}$:

$$(\hat{\mathcal{P}}) \qquad \begin{cases} \hat{A}\hat{u} = \hat{f} & \text{in } \hat{\Omega} \\ +b.c. & \text{on } \partial\hat{\Omega}. \end{cases}$$

$(\hat{\mathcal{P}})$ has to be chosen in such a way that the following requirement concerning its solution $\hat{u}$ is satisfied: *the restriction* $\hat{u}\,|_\Omega$ *gives a solution* to $(\mathcal{P})$. Thus instead of $(\mathcal{P})$, we solve $(\hat{\mathcal{P}})$. The gain is obvious: since $\hat{\Omega}$ has a simple shape, there is no problem to construct specific regular partitions of $\hat{\Omega}$ into finite elements. Moreover, such partitions generate special stiffness matrices enabling us to use fast solvers to get the solution of the corresponding system. Below we present two ways, how the auxiliary problem on $\hat{\Omega}$, fulfilling the above mentioned requirement can be defined.

3.1. DUALITY APPROACH

Here we restrict ourselves to a homogeneous Dirichlet boundary value problem for the Laplace equation, whose weak form reads as follows:

$$(\mathcal{P}) \qquad \begin{cases} \text{Find } u \in H_0^1(\Omega) \text{ such that} \\ \int_\Omega \operatorname{grad} u \cdot \operatorname{grad} \varphi \, dx = \int_\Omega f\varphi \, dx \quad \forall \varphi \in H_0^1(\Omega), \end{cases}$$

with $f \in L^2(\Omega)$ given. The symbol $H^1(\Omega)$ stands for the space of functions, which together with their first derivatives are square integrable in Ω, i.e. they are elements of $L^2(\Omega)$. The space $H_0^1(\Omega)$ contains all functions from $H^1(\Omega)$ vanishing on $\partial\Omega$.

Let $\hat{\Omega} \supset \overline{\Omega}$ be a fictitious domain and denote by $\hat{V} = H_0^1(\hat{\Omega})$. Problem $(\mathcal{P})$ will be now replaced by the following *saddle-point type formulation*:

$$(\hat{\mathcal{P}}) \qquad \begin{cases} \text{Find } (\hat{u}, \lambda) \in \hat{V} \times \Lambda \text{ such that} \\ \int_{\hat{\Omega}} \operatorname{grad} \hat{u} \cdot \operatorname{grad} \varphi \, dx = \int_{\hat{\Omega}} \tilde{f}\varphi \, dx + < \lambda, \varphi > \quad \forall \varphi \in \hat{V} \\ < \mu, \hat{u} >= 0 \quad \forall \mu \in \Lambda, \end{cases}$$

where Λ is a space of Lagrange multipliers being in duality denoted by $<,>$ with $\hat{V}$. The symbol $\tilde{f}: \hat{\Omega} \to \mathbb{R}^1$, $\tilde{f} \in L^2(\hat{\Omega})$ is an extension of f from Ω onto $\hat{\Omega}$.

Next we shall consider two types of Λ:

(i) Λ is the dual to the trace space on $\partial\Omega$ of functions from $\hat{V}$, i.e. $\Lambda = H^{-1/2}(\partial\Omega)$ (the boundary Lagrange multiplier technique);
(ii) Λ is the dual to the space of all restrictions to $\Xi \equiv \hat{\Omega} \setminus \overline{\Omega}$ of functions from $\hat{V}$, i.e. $\Lambda = (\hat{V}\,|_\Xi)'$ (the distributed Lagrange multiplier technique).

It is easy to prove that with Λ given by (i) or (ii), problem $(\hat{\mathcal{P}})$ has a unique solution $(\hat{u}, \lambda)$ and the restriction $\hat{u}\,|_\Omega$ solves $(\mathcal{P})$. For details we refer to Haslinger & Klarbring (1995), Tomas (1997) and Haslinger, Maître & Tomas (1998).

REMARK 3.1 The Lagrange multiplier λ can be interpreted as a fictitious force which has to be applied in ω in order to get $u \equiv 0$ in ω. The set ω is equal to $\partial\Omega$, Ξ for Λ given by (i), (ii), respectively.

Let $\hat{V}_h$, Λ_H be finite dimensional subspaces of $\hat{V}$ and Λ, respectively. Then the *approximation* of $(\hat{\mathcal{P}})$ reads as follows:

$$(\hat{\mathcal{P}})_{hH} \qquad \begin{cases} \text{Find } (\hat{u}_h, \lambda_H) \in \hat{V}_h \times \Lambda_H \text{ such that} \\ \int_{\hat{\Omega}} \operatorname{grad} \hat{u}_h \cdot \operatorname{grad} \varphi_h \, dx = \int_{\hat{\Omega}} \tilde{f}\varphi_h \, dx + [\lambda_H, \varphi_h] \\ \qquad\qquad\qquad\qquad\qquad\qquad \forall \varphi_h \in \hat{V}_h; \\ [\mu_H, \hat{u}_h] = 0 \quad \forall \mu_H \in \Lambda_H, \end{cases}$$

where $[\,,\,]$ stands for the duality between $\hat{V}_h$ and Λ_H. Below we briefly describe the construction of $\hat{V}_h$ and Λ_H which will be used in Section 5. We restrict ourselves to the plane case when $\hat{\Omega}$ is a sufficiently large rectangle containing $\overline{\Omega}$.

Let $\hat{\mathcal{R}}_h$ be a rectangulation of $\hat{\Omega}$, i.e. $\overline{\hat{\Omega}}$ is the union of a finite number of rectangles K_i, $i = 1, \dots, q$ having in common a whole side or a vertex at most. Moreover, the diameter of each $K_i \in \hat{\mathcal{R}}_h$ does not exceed h. Then $\hat{V}_h$ is the space of all *continuous piecewise bilinear functions* over $\hat{\mathcal{R}}_h$ vanishing on $\partial\hat{\Omega}$. The definition of Λ_H depends on the type of Λ.

Let Λ be given by (i). By Ω_H we denote a *polygonal approximation* of Ω, i.e. $\partial\Omega_H$ is the union of a finite number of segments $\overline{A_i A_{i+1}}$, $i = 1, \dots, m$ ($A_{m+1} \equiv A_1$), where A_i are the vertices of Ω_H placed on $\partial\Omega$ and such that the length of any $\overline{A_i A_{i+1}}$ is less or equal than H. Then Λ_H is the space of functions defined on $\partial\Omega_H$ which are constant on any $\overline{A_i A_{i+1}}$, $i = 1 \dots, m$. The duality pairing $[\,,\,]$ is represented by the $L^2(\partial\Omega_H)$-scalar product.

Let Λ be given by (ii). Denote by $\hat{\mathcal{R}}_H$ another rectangulation of $\hat{\Omega}$ such that $\hat{\mathcal{R}}_H \subseteq \hat{\mathcal{R}}_h$, i.e. any element of $\hat{\mathcal{R}}_H$ can be written as the union of a finite number of elements from $\hat{\mathcal{R}}_h$. By $\hat{V}_H$ we denote the space of all continuous, piecewise bilinear functions over $\hat{\mathcal{R}}_H$ vanishing on $\partial\hat{\Omega}$. Then $\Lambda_H \overset{\text{def}}{\equiv} \hat{V}_H|_\Xi$ and the duality pairing $[\,,\,]$ is realized by the $L^2(\Xi)$-scalar product.

With these choices of $\hat{V}_h$ and Λ_H, problem $\left(\hat{\mathcal{P}}\right)_{hH}$ has a unique solution $(\hat{u}_h, \lambda_H)$ provided that the ratio H/h is sufficiently large in the case of Λ given by (i). If Λ is given by (ii) no such restriction is needed. For details we refer again to Haslinger & Klarbring (1995) and Haslinger, Maître & Tomas (1998). The matrix representation of $\left(\hat{\mathcal{P}}\right)_{hH}$ leads to the following typical saddle-point formulation:

$$\begin{pmatrix} \mathbb{A} & \mathbb{B}^T \\ \mathbb{B} & \mathbb{O} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{F} \\ \mathbf{0} \end{pmatrix}, \tag{3.1}$$

where $\mathbb{A}$ is the *stiffness matrix*, $\mathbb{B}$ is the so-called *kinematic transformation* matrix, $\mathbf{F}$ is the load vector and $\mathbf{u}$, $\boldsymbol{\lambda}$ are the vectors of the nodal values of $\hat{u}_h$, λ_H. Let us recall that the elements b_{ij} of the matrix $\mathbb{B}$ are given by

$$b_{ij} = \int_{\overline{A_i A_{i+1}}} \varphi_j \, ds, \quad i = 1, \dots, m, \quad j = 1, \dots, n$$

in the case of the boundary Lagrange multipliers or by

$$b_{ij} = \int_\Xi \psi_i \varphi_j \, dx, \quad i = 1, \dots, \overline{m}, \quad j = 1, \dots, n$$

in the case of the distributed Lagrange multipliers. Here $\{\varphi_j\}_{j=1}^n$, $\{\psi_i\}_{i=1}^{\overline{m}}$ are basis functions of $\hat{V}_h$, $\hat{V}_H|_\Xi$, respectively. It is important to notice that the only information on the geometry of Ω is encoded in the matrix $\mathbb{B}$ (eventually the load vector $\mathbf{F}$) but *not in the stiffness matrix* $\mathbb{A}$.

3.2. OPTIMAL CONTROL APPROACH

This approach will be used for the solution of the Neumann problem:

$$\begin{cases} -\triangle u + u = f \ \text{ in } \ \Omega, \quad f \in L^2(\Omega) \\ \qquad \frac{\partial u}{\partial \nu} = g \ \text{ on } \ \partial\Omega, \quad g \in L^2(\partial\Omega), \end{cases} \tag{3.2}$$

or in the weak form:

$$\begin{cases} \text{Find } u \in H^1(\Omega) \text{ such that} \\ a_\Omega(u, \varphi) =< f, \varphi > \quad \forall \varphi \in H^1(\Omega), \end{cases} \tag{3.3}$$

where

$$\begin{aligned} a_\Omega(u, \varphi) &\equiv \int_\Omega (\operatorname{grad} u \cdot \operatorname{grad} \varphi + u\varphi)\, dx, \\ < f, \varphi > &\equiv \int_\Omega f\varphi\, dx + \int_{\partial\Omega} g\varphi\, ds. \end{aligned}$$

Again, let $\hat{V} = H_0^1(\hat{\Omega})$. Instead of (3.3) we shall consider the following optimal control problem:

$$\begin{cases} \text{Find } \overline{v} \in H^1(\Omega) \text{ such that} \\ J(\overline{v}) = \min\limits_{v \in H^1(\Omega)} J(v), \end{cases} \tag{3.4}$$

where $J(v) = \frac{1}{2}\int_{\hat{\Omega}} |\operatorname{grad} \hat{y}(v)|^2\, dx$ with $\hat{y} \equiv \hat{y}(v) \in \hat{V}$ being the solution of *the state problem* ($v \in H^1(\Omega)$):

$$\left(\hat{\mathcal{P}}(v)\right) \ \int_{\hat{\Omega}} \operatorname{grad} \hat{y} \cdot \operatorname{grad} \varphi\, dx = a_\Omega(v, \varphi\,|_\Omega) - < f, \varphi\,|_\Omega > \quad \forall \varphi \in \hat{V}.$$

From (3.4) and the definition of $\left(\hat{\mathcal{P}}(v)\right)$ we see that any $\overline{v}$ realizing the absolute minimum of J in $H^1(\Omega)$ solves (3.3) at the same time.

The finite element approximation of (3.4) reads as follows:

$$\begin{cases} \text{Find } \overline{v}_H \in \Lambda_H \text{ such that} \\ J(\overline{v}_H) = \min\limits_{v_H \in \Lambda_H} J(v_H), \end{cases} \tag{3.5}$$

where $J(v_H) = \frac{1}{2}\int_{\hat{\Omega}} |\operatorname{grad} \hat{y}_h(v_H)|^2\, dx$ with $\hat{y}_h \equiv \hat{y}_h(v_H) \in \hat{V}_h$ being the solution of

$$\begin{cases} \int_{\hat{\Omega}} \operatorname{grad} \hat{y}_h \cdot \operatorname{grad} \varphi_h\, dx = a_\Omega(v_H, \varphi_h\,|_\Omega) - < f, \varphi_h\,|_\Omega > \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall \varphi_h \in \hat{V}_h, \end{cases} \tag{3.6}$$

where $\Lambda_H \equiv \hat{V}_H\,|_\Omega$ and $\hat{V}_h$, $\hat{V}_H$ are the same as before.

The algebraic formulation of (3.5) leads to the following quadratic programming problem:

$$\begin{cases} \text{Find } \mathbf{v}^* \in \mathbb{R}^d \text{ such that} \\ \mathcal{J}(\mathbf{v}^*) = \min\limits_{\mathbf{v}\in\mathbb{R}^d} \mathcal{J}(\mathbf{v}), \end{cases} \tag{3.7}$$

where

$$\mathcal{J}(\mathbf{v}) = \frac{1}{2}\big(\mathbb{A}\,\mathbf{y}(\mathbf{v}), \mathbf{y}(\mathbf{v})\big)_{\mathbb{R}^n},$$

$d =$ diam Λ_H and $\mathbf{y}(\mathbf{v}) \in \mathbb{R}^n$ is the solution of

$$\mathbb{A}\,\mathbf{y}(\mathbf{v}) = \mathbb{B}\,\mathbf{v} - \tilde{\mathbf{F}}. \tag{3.8}$$

Here $\mathbb{A}$ denotes the stiffness matrix, $\tilde{\mathbf{F}}$ is the vector representation of $< f, \varphi_h |_\Omega >$ and the elements of $\mathbb{B}$ are given by

$$b_{ij} = a_\Omega(\psi_j, \varphi_i),$$

where $\{\varphi_i\}_{i=1}^n$, $\{\psi_j\}_{j=1}^d$ are basis functions of $\hat{V}_h$, Λ_H, respectively.

3.3. FICTITIOUS DOMAIN APPROACHES IN SHAPE OPTIMIZATION

Let $(\mathbf{P})$ be an optimal shape design problem whose abstract form has been introduced in Section 2. To make its realization more efficient, we shall use FDS's as a tool for the numerical realization of state problems. Thus we propose the following approximation of $(\mathbf{P})$:

$$(\hat{\mathbf{P}}_h) \qquad \begin{cases} \text{Find } \Omega_h^* \in \mathcal{O}_h \text{ such that} \\ J\big(\Omega_h^*, \hat{u}_h(\Omega_h^*)\,|_{\Omega_h^*}\big) = \min\limits_{\Omega_h\in\mathcal{O}_h} J\big(\Omega_h, \hat{u}_h(\Omega_h)\,|_{\Omega_h}\big), \end{cases}$$

where $\mathcal{O}_h$ is an approximation of $\mathcal{O}$ and $\hat{u}_h(\Omega_h)$ is an approximation of $u(\Omega)$, realized by one of the previous fictitious domain techniques, used on $\Omega \equiv \Omega_h$. Here we use the symbol Ω_h as the argument of $\hat{u}_h$ to point out that our fictitious domain solution *depends* on the geometry of Ω_h.

The advantage of this approach is readily seen: the state problem is always solved on the *same* domain $\hat{\Omega}$, using the *same*, *fixed*, regular partition of $\hat{\Omega}$ into finite elements. Let us recall that in the case of the Lagrange multiplier approaches we arrive at the following linear algebraic system:

$$(\hat{\mathcal{P}}(\boldsymbol{\alpha})) \qquad \begin{cases} \text{Find } \big(\mathbf{u}(\boldsymbol{\alpha}), \boldsymbol{\lambda}(\boldsymbol{\alpha})\big) \in \mathbb{R}^n \times \mathbb{R}^d \text{ such that} \\ \begin{pmatrix} \mathbb{A} & \mathbb{B}^T(\boldsymbol{\alpha}) \\ \mathbb{B}(\boldsymbol{\alpha}) & \mathbb{O} \end{pmatrix} \begin{pmatrix} \mathbf{u}(\boldsymbol{\alpha}) \\ \boldsymbol{\lambda}(\boldsymbol{\alpha}) \end{pmatrix} = \begin{pmatrix} \mathbf{F}(\boldsymbol{\alpha}) \\ \mathbf{0} \end{pmatrix}. \end{cases}$$

As we have already mentioned, the only information on the geometry of Ω_h expressed through $\boldsymbol{\alpha}$ is encoded in $\mathbb{B}$ and $\mathbf{F}$. Just the fact that $\mathbb{A}$ does not depend on $\boldsymbol{\alpha}$ enables us to solve $\big(\hat{\mathcal{P}}(\boldsymbol{\alpha})\big)$ efficiently. The similar statement holds true when the optimal control approach is used. Thus the matrix $\mathbb{A}$ can be *computed once* and *factorized for ever*.

As mentioned in the introduction, the use of FDS's in shape optimization gives rise to certain difficulties which exclude the use of classical gradient type minimization methods. We start with the *sensitivity analysis* or how the solution $\mathbf{u}$ of $\big(\hat{\mathcal{P}}(\boldsymbol{\alpha})\big)$ depends on variations of $\boldsymbol{\alpha}$.

When the boundary Lagrange multipliers are used, the elements $b_{ij}(\boldsymbol{\alpha})$ of $\mathbb{B}(\boldsymbol{\alpha})$ are given by

$$b_{ij}(\boldsymbol{\alpha}) = \int_{\overline{A_i A_{i+1}}} \varphi_j \, ds,$$

where $\overline{A_i A_{i+1}}$ is the side of a polygonal domain Ω_H being an approximation of $\Omega_h \in \mathcal{O}_h$ ($\Omega_H \equiv \Omega_h$ when Ω_h itself is polygonal). It is very easy to verify that if $\overline{A_i A_{i+1}}$ has a non-empty intersection with an interelement boundary between two adjacent rectangles belonging to $\hat{\mathcal{R}}_h$ whose 1-dimensional Lebesgue measure is positive, then the mapping

$$\boldsymbol{\alpha} \mapsto b_{ij}(\boldsymbol{\alpha})$$

is not continuously differentiable (for details we refer to Haslinger & Neittaanmäki 1996; Daňková & Haslinger 1996; Haslinger & Klarbring 1995). Thus one can not expect that the mapping $\boldsymbol{\alpha} \mapsto \mathbf{u}(\boldsymbol{\alpha})$ is differentiable. The same holds when the optimal control approach is used. In this case the elements of $\mathbb{B}(\boldsymbol{\alpha})$ are given by

$$\int_{\Omega_h} \operatorname{grad} \psi_i \cdot \operatorname{grad} \varphi_j \, dx$$

containing the discontinuous integrand. In both cases however, the mapping $\boldsymbol{\alpha} \mapsto \mathbf{u}(\boldsymbol{\alpha})$ is continuous so that the whole optimization process becomes *non-smooth*, in general. When the distributed Lagrange multipliers are used, the situation is more involved due to the so called *locking effect*: let $\hat{\mathcal{R}}_h \equiv \hat{\mathcal{R}}_H$, i.e. the same rectangulation is used for the construction of the spaces, approximating the solution and the Lagrange multiplier. Then from

$$\int_{\Xi_h} \mu_h \hat{u}_h \, dx = 0 \quad \forall \mu_h \in \Lambda_h = \hat{V}_h / \Xi_h, \qquad \Xi_h = \hat{\Omega} \backslash \overline{\Omega}_h$$

it follows that $\hat{u}_h \equiv 0$ not only in Ξ_h, but due to the fact that $\hat{u}_h$ is piecewise bilinear, this solution is identically equal to zero in a larger set Ξ_h', where

$$\Xi_h' = \bigcup \big\{ T; \quad \operatorname{int} T \cap \Xi_h \neq \emptyset \big\}$$

is the union of all rectangles from $\hat{\mathcal{R}}_h$, whose interior has a non-empty intersection with Ξ_h. Now, let $\boldsymbol{\alpha}$ be such that its small perturbations do not change the set Ξ_h'.

Then the solution $\mathbf{u}$ before and after the modification of such $\boldsymbol{\alpha}$ remains the same (see Haslinger, Maître & Tomas 1998). This means that the mapping $\boldsymbol{\alpha} \mapsto \mathbf{u}(\boldsymbol{\alpha})$ has a *finite range* and therefore it is even *discontinuous*. If the cost functional does not depend explicitly on Ω, for instance, when the identification is made through a fixed target set, then the optimization problem is of a *discrete type*.

From what it has been said it follows that local optimization methods can hardly be used.

4. Algorithms

Evolutionary algorithms are probabilistic algorithms which solve the global optimitization problems through modelling of organic evolution. Among them, Genetic Algorithm (GA), Breeder Genetic Algorithm (BGA), and Modified Controlled Random Search (MCRS) are representants we used to solve the problems introduced in Section 3. As the classical GA is well known, only BGA and MCRS will be mentioned.

4.1. OPTIMIZATION PROBLEM

The global optimization problem can be stated as follows

$$\mathbf{x}^* = \arg\min \left\{ f\left(\tilde{\mathbf{x}}\right), \tilde{\mathbf{x}} \in \mathcal{X} \right\}, \tag{4.1}$$

where $\mathcal{X}$ is an admissible search space, $f : \mathcal{X} \subseteq \mathcal{X}_1 \times \cdots \times \mathcal{X}_n \to \mathbb{R}$ is an *objective function* and $f(\mathbf{x}^*)$ is a *global minimum*. In constrained problems the set of all feasible points is defined as follows:

$$\mathcal{X} = \left\{ \tilde{\mathbf{x}} \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_n, \quad g_j\left(\tilde{\mathbf{x}}\right) \geqslant 0 \quad \forall j \in \{1, \ldots, m\} \right\},$$

where $g_j : \mathcal{X}_1 \times \cdots \times \mathcal{X}_n \to \mathbb{R}$ are inequality constraints. Usually $\mathcal{X}_i$ are in the form of box constraints, i.e $\mathcal{X}_i = \langle a_i, b_i \rangle \subset \mathbb{R}$.

The restriction to minimization problems is without loss of generality, because

$$-\max\left(f\left(\mathbf{x}\right)\right) = \min\left(-f\left(\mathbf{x}\right)\right).$$

4.2. BREEDER GENETIC ALGORITHM – BGA

The Breeder Genetic Algorithm is based on the same concept as a typical GA. The name of BGA is derived from the selection mechanism which implements some type of breeding. At each generation of $\mathcal{N}$ elements, the best $\mathcal{T} \cdot \mathcal{N}$ elements are selected, where $\mathcal{T}$ is the so-called *truncation rate*, usually $\mathcal{T} \in \langle 0.1, 0.5 \rangle$. Individuals from this selection are randomly chosen to mate with the same probability.

The main difference between the GA and BGA is that solutions are represented as real numbers instead of binary strings. This difference in representation of values

leads to the need of introducing new recombination and mutation operators. We will now briefly describe these operators.

In what follows $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$ denote the parent strings, and $\mathbf{z} = (z_1, \ldots, z_n)$ denotes the offspring, where all $x_i, y_i, z_i \in \mathcal{X}_i \subset \mathbb{R}$ and $i \in \{1, \ldots, n\}$.

4.2.1. *Recombination*

There are three recombination operators.
Discrete recombination – DR

$$z_i = x_i \quad or \quad y_i,$$

where x_i or y_i are chosen with probability 0.5.
Extended line recombination – ELR

$$z_i = x_i + \alpha(y_i - x_i),$$

where α is chosen uniform randomly in the range $\langle -\delta, 1 + \delta \rangle$.
Extended intermediate recombination – EIR

$$z_i = x_i + \alpha_i(y_i - x_i),$$

where α_i is chosen uniform randomly in the range $\langle -\delta, 1 + \delta \rangle$. In both cases usually $\delta = 0.25$.

Looking at the strings of values as the points in a vector space, we can describe the geometrical effect of the recombination operators. DR only mixes the values of parent points and generates vertices of the hypercube defined by these points. ELR and EIR generate new values by linear combination of the corresponding parent values. ELR generates points on the line between the parents, while EIR generates points inside the hypercube defined by the parents. Both ELR and EIR work with an *extent* δ. Thus the hypercube where the generated points can be found is larger than the hypercube generated by parent points.

4.2.2. *Mutation*

Each variable x_i is selected for mutation with probability p_m, usually $p_m = 1/n$. For each variable x_i we define $range_i$. When $x_i \in \langle a_i, b_i \rangle$, we usually set $range_i = 0.1(b_i - a_i)$.
Discrete mutation scheme – DMS

$$z_i = x_i \pm range_i \sum_{j=1}^{k-1} \alpha_j \cdot 2^{-j},$$

where $\alpha_j \in \{0, 1\}$ are chosen to be equal to 1 with probability $1/k$, k is an integer called *precision constant*.
Continuous mutation scheme – CMS

$$z_i = x_i \pm range_i \cdot 2^{-k\alpha}$$

where α is chosen uniformly from range $\langle 0, 1\rangle$ and k is again the *precision constant*.

From the geometrical point of view both mutations can generate any point in the hypercube with the center in **x** defined by the points $(x_1 - range_1, \ldots, x_n - range_n)$ and $(x_1 + range_1, \ldots, x_n + range_n)$. But with high probability, points in the neighborhood of **x** are generated.

For more details on the BGA see Mühlenbein & Schlierkamp-Voosen (1992). The predictive model of BGA is described in Mühlenbein & Schlierkamp-Voosen (1993).

4.3. MCRS ALGORITHM

The MCRS (Modified Controlled Random Search) algorithm (Křivý & Tvrdík 1995, 1996) is based on old ideas of the simplex method (Nelder & Mead 1964) and the controlled random search (Price 1976). The MCRS algorithm starts with a population P of N points taken at random in $\mathcal{X}$. A new trial point **x** is generated from a simplex S (a set of $n + 1$ linearly independent points of a population P in $\mathcal{X}$) by the relation

$$\mathbf{z} = \mathbf{g} - Y(\mathbf{x} - \mathbf{g}), \tag{4.2}$$

where **x** is one (randomly chosen) vertex of the simplex S, **g** is the center of gravity of the remaining n vertices of the simplex and Y is a multiplicative factor. The point **z** may be considered as resulting from the reflection of the point **x** with respect to **g**.

The principal modification of the original Price's reflection consists in randomizing the multiplicative factor Y. Instead of Y being constant, a random variable is used in the MCRS algorithm. Several distributions of Y have been tested on fourteen hard problems of estimation of non-linear regression parameters. It was found that the best results were obtained with Y distributed uniformly in $\langle 0, \alpha)$ with α ranging from 4 to 8, see Křivý & Tvrdík (1995).

Considering the procedure $Reflection$ which can be formally written as

procedure $Reflection(P,$ **var z**$)$;
repeat
  $S :=$ set of $(n + 1)$ points selected from P at random;
  z := **g** $- Y($**x** $-$ **g**$)$;
until z $\in \mathcal{X}$.

then the description of the MCRS algorithm is very simple:

procedure MCRS;
begin $P :=$ population of N randomly generated points in $\mathcal{X}$,
  repeat
    $Reflection(P, \mathbf{z})$;
    if $f(\mathbf{z}) < f(\mathbf{z}_{max})$ **then** $\mathbf{z}_{max} := \mathbf{z}$;
  until stopping criterion is true;
end {MCRS};

$\mathbf{z}_{max}$ is the point with the largest function value from the N points currently stored.

No particular stopping criterion is defined. However, in the most of optimization problems the stopping condition is expressed as

$$f(\mathbf{z}_{max}) - f(\mathbf{z}_{min}) \leqslant \varepsilon, \tag{4.3}$$

where $\mathbf{z}_{min}$ is the point with the smallest value of the objective function among all the N points of population P held in the memory and ε is an input parameter.

Thus, the algorithm has only three input tuning parameters:

- the number of points N in the population;
- the value of α defining the range of Y;
- the value of ε for the stopping condition.

A right setting of the tuning parameters is dependent on the nature of optimization problem to be solved. It is obvious that the higher values of N and α, the more thorough is the search.

The MCRS algorithm is similar to other evolutionary algorithms in several aspects. The reflection plays the role of the recombination like the crossover in the genetic algorithm. The probability of selection in the MCRS is not proportional to the fitness of a selected individual but due to discarding the worst individual in a population and replacing it by a better new point there is some self-adaptation tendency in the population. However, mutation is not explicitly included into the MCRS. An attempt to include the operator of mutation into the algorithm was made. It leads to the so-called evolutionary search algorithms (Křivý & Tvrdík 1996, 1997), which were found to be more reliable in searching the *true* global minimum but with lower convergence rate.

No proof of convergence of the MCRS algorithm has been done yet. We suppose that the MCRS algorithm is at least as good as the uniform random search for which the convergence with probability one was proven, e.g., Bäck (1992) pp. 48–49. It was found experimentally on several problems that the convergence rate of the MCRS is much higher than the convergence rate of the uniform random search.

5. Numerical Examples

In this section we present results of three model optimal shape design problems using FDS's as a tool for the numerical realization of state problems. The minimization itself is performed by the GA, BGA and MCRS algorithm.

In all examples the fictitious domain $\hat{\Omega}$ is the rectangle $\hat{\Omega} = (0, Lx) \times (0, Ly)$, which is divided into small squares with the step h, defining the rectangulation $\hat{\mathcal{R}}_h$ of $\hat{\Omega}$. The approximation $\mathcal{O}_h$ of $\mathcal{O}$ is realized by domains Ω_h whose boundaries are piecewise second degree Bezier curves. We define n_b as the number of segments made of these curves and n_c as the number of control points. An example of such a boundary is shown in Figure 1. Each Bezier curve is defined by the initial, end and

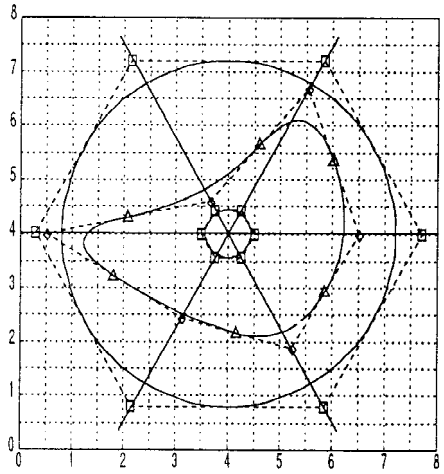

Figure 1.

the control point. Initial and end points are denoted by small triangles and control points by circles. Each control point moves on line between small squares, thus the position of the control point can be expressed by a real number from the interval $< C_0, C_1 >$, where C_0, C_1 are the minimal, maximal, respectively distance of the control point from the center $(Lx/2, Ly/2)$ of the fictitious domain $\hat{\Omega}$.

The admissible family of domains $\mathcal{O}$ is the subset of the following set

$$\mathcal{M}(C_0, C_1) = \{\Omega \subset \hat{\Omega} \mid \quad C_0 \leqslant \|X - (Lx/2, Ly/2)\| \leqslant C_1 \quad \forall X \in \partial\Omega\}.$$

To solve the discrete state problem

$$\mathbb{A}\,\mathbf{x} = \mathbf{b},$$

we use the conjugate gradient method. The stopping criteria for this method is stated as follows:

$$\|\mathbf{r}\|^2 \leqslant \|\mathbf{b}\|^2 \varepsilon_{CG},$$

where $\mathbf{r}$ is the residuum and $\varepsilon_{CG} > 0$ is a small real constant.

EXAMPLE 1 (the boundary Lagrange multiplier technique).

Let $Lx = Ly = 8$, $h = 1/2$, $\varepsilon_{CG} = 10^{-4}$,

$$\mathcal{O} = \left\{\Omega \in \mathcal{M}(1.4, 2.9) \mid \quad \text{meas } \Omega = 4\pi\right\}$$

be the admissible family of domains containing the origin inside and satisfying the constant volume constraint. On any $\Omega \in \mathcal{O}$ we consider the state problem

$$\left(\mathcal{P}(\Omega)\right) \qquad \begin{cases} -\triangle u = 4 \ \text{ in } \ \Omega, \quad \Omega \in \mathcal{O} \\ \quad u = 0 \ \text{ on } \ \partial\Omega. \end{cases}$$

Further let

$$J_1(\Omega) = -4 \int_\Omega u(\Omega)\, dx$$

be the cost functional. We define the optimal shape design problem

$$\left(\mathbf{P1}\right) \qquad \begin{cases} \text{Find } \Omega^* \in \mathcal{O} \text{ such that} \\ J_1\left(\Omega^*\right) \leqslant J_1\left(\Omega\right) \quad \forall \Omega \in \mathcal{O}. \end{cases}$$

Problem $\left(\mathbf{P1}\right)$ is one of the classical shape optimization problems: namely to find a shape of the cross section of a shaft made of a homogenous material in order to maximize its torsional rigidity. The result is known: the optimal shape is realized by a circle.

$\mathcal{O}_h$ is realized by Bezier curves with $n_c = 6$. Thus the nodal vector representing the boundary Lagrange multiplier has six components. The optimal shape is shown in Figure 2 (realized by MCRS after 1000 evaluations of the cost functional). The control points are the object of minimization.

EXAMPLE 2 (the distributed Lagrange multiplier technique).
The admissible family of domains is defined by

$$\mathcal{O} = \left\{\Omega \in \mathcal{M}(0.5, 1.5) \mid \quad \text{meas } \Omega = 1.5342\right\}$$

and $Lx = Ly = 3$, $h = 3/16$, $\varepsilon_{CG} = 10^{-5}$. The state problem is defined by

$$\left(\mathcal{P}(\Omega)\right) \qquad \begin{cases} -\triangle u = f \ \text{ in } \ \Omega, \quad \Omega \in \mathcal{O} \\ \quad u = 0 \ \text{ on } \ \partial\Omega, \end{cases}$$

where

$$f = -\triangle u_z$$

with

$$u_z = \left(x - \frac{Lx}{2} + c\right)\left(\alpha(y) - x\right)\left(y - \frac{Ly}{2} + c\right)\left(\frac{Ly}{2} + c - y\right),$$

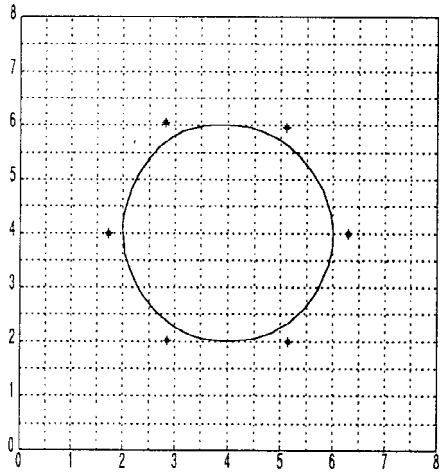

Figure 2.

$$\alpha(y) = \frac{3}{8} sin\left(\frac{\pi \overline{y}}{2c}\right) + \frac{Lx}{2} + c,$$

$$\overline{y} = y - \frac{Ly}{2} + c, \quad c = 0.5625.$$

It can be easily verified that u_z is the solution of $\big(\mathcal{P}(\tilde{\Omega})\big)$, where $\tilde{\Omega} \in \mathcal{O}$ is surrounded by the zero level set of u_z (curved rectangle whose one curved side is represented by the graph of the function $x = \alpha(y)$). We define the optimal shape design problem as follows:

$$(\mathbf{P2}) \qquad \begin{cases} \text{Find } \Omega^* \in \mathcal{O} \text{ such that} \\ J_2\big(\Omega^*\big) \leqslant J_2\big(\Omega\big) \quad \forall \Omega \in \mathcal{O}, \end{cases}$$

where

$$J_2(\Omega) = \int_\Omega (u(\Omega) - u_z)^2 dx.$$

Thus $\tilde{\Omega}$ is one of solutions to $(\mathbf{P2})$. Distributed Lagrange multipliers are used for the realization of the state problem. As before, piecewise bilinear functions are used for constructing $\hat{V}_h$ and Λ_H with $H = 2h = 3/8$, i.e. the partition $\hat{\mathcal{R}}_H$ of $\hat{\Omega}$ into squares defining the space Λ_H is two times coarser as this one used in the definition of $\hat{V}_h$.

$\mathcal{O}_h$ is realized by Bezier curves with $n_c = 4$. In this case the position of both control and initial points is optimized. An example of the optimal shape is shown in Figure 3 (realized by GA with exponential scaling and uniform crossover after 1000 evaluations of the cost functional).

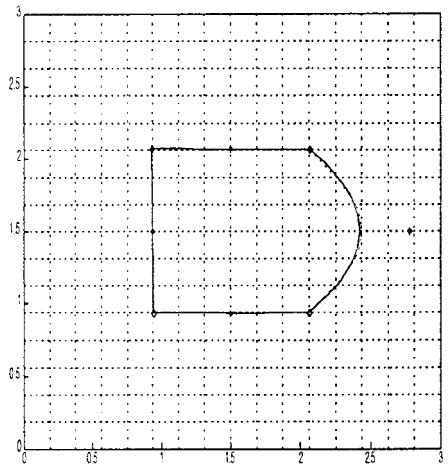

Figure 3.

As mentioned in Section 3, the distributed Lagrange multiplier approach exhibits the locking effect yielding a discrete type optimization problem. We shall illustrate this phenomenon by drawing the graph of the cost functional J_2 assumed to be a function of two design variables $\{3, 4\}$, keeping the rest fixed. We take $\hat{\mathcal{R}}_h = \hat{\mathcal{R}}_H$.

The Figure 4 displays the graph. The consequence of the locking effect described in Section 3 is seen well: the function is stairwise.

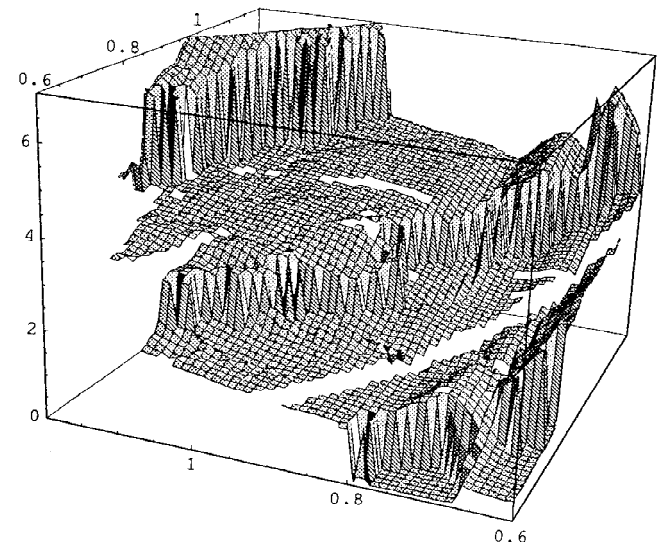

Figure 4.

EXAMPLE 3 (the optimal control approach).
Let $Lx = Ly = 8$, $h = 1/2$, $\varepsilon_{CG} = 10^{-5}$,

$$\mathcal{O} = \{\Omega \in \mathcal{M}(1.0, 2.5) \mid \quad \text{meas } \Omega = 2\pi\}.$$

On any $\Omega \in \mathcal{O}$ we consider the Neumann problem:

$$(\mathcal{P}(\Omega)) \qquad \begin{cases} -\triangle u(\Omega) + u(\Omega) = f \ \text{ in } \ \Omega, \quad \Omega \in \mathcal{O}, \\ \dfrac{\partial u(\Omega)}{\partial \nu} = g \ \text{ on } \ \partial\Omega, \end{cases}$$

where

$$f = -\triangle u_z + u_z, \quad g = \frac{\partial u_z}{\partial \nu},$$

$$u_z = 4 - \left(x - \frac{Lx}{2}\right)^2 - 4\left(y - \frac{Ly}{2}\right)^2$$

and the normal derivative of u_z is assumed on $\partial\tilde{\Omega}$ given by the ellipse

$$\frac{\left(x - \frac{Lx}{2}\right)^2}{4} + \left(y - \frac{Ly}{2}\right)^2 = 1. \tag{5.1}$$

We define the optimal shape design problem

$$(\mathbf{P3}) \qquad \begin{cases} \text{Find } \Omega^* \in \mathcal{O} \text{ such that} \\ J_3(\Omega^*) \leqslant J_3(\Omega) \quad \forall \Omega \in \mathcal{O}, \end{cases}$$

where

$$J_3(\Omega) = \int_\Omega (u(\Omega) - u_z)^2 \, dx.$$

It is easy to see that one of solutions to $(\mathbf{P3})$ is the domain $\tilde{\Omega}$ being the interior of (5.1). State problem $(\mathcal{P}(\Omega))$ was solved by using the optimal control approach. As before, piecewise bilinear functions were used for constructing the spaces $\hat{V}_h$, Λ_H with $H = h = 1/2$, i.e. we use the same rectangulation for both spaces.

In this case $n_c = 8$. An example of the optimal shape is shown in Figure 5 (realized by BGA after 1000 evaluations of the cost functional) and the control points are the object of minimization.

5.1. COMPARISON OF ALGORITHMS

In this section we describe and compare the results obtained. We run the following optimization algorithms:

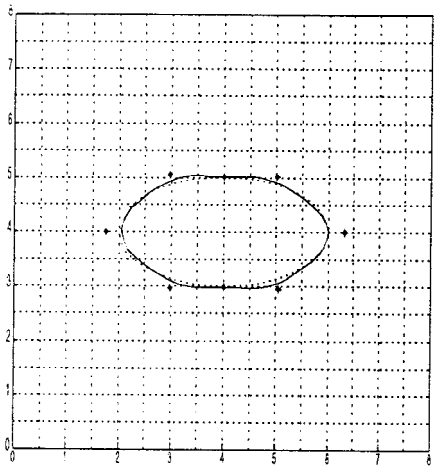

Figure 5.

MCRS – modified controlled random search algorithm (described in Section 4.3);

BGA – breeder genetic algorithm (described in Section 4.2);

GA_{bu} – GA with truncation selection and uniform crossover;

GA_{b2} – GA with truncation selection and one-point crossover;

GA_{eu} – GA with exponential selection and uniform crossover;

GA_{e2} – GA with exponential selection and one-point crossover.

Optimization was run several times for each example and each type of optimization algorithm. The population size for MCRS, all the variants of genetic algorithms was equal to n^2, 20, respectively, where n is the number of optimized variables being the argument of the cost function. The value of α defining the range of Y in the MCRS was equal to 4. The parameters for BGA were chosen as follows:

crossover probability $p_c = 0.8$,
mutation probability $p_m = 1/n$,
crossover extent $\delta = 0.25$,
precision constant $k = 20$,
$range_i = 1.0$.

In the case of the classical GA the following parameters were used:

crossover probability $p_c = 0.6$, mutation probability $p_m = 0.004$,
truncation rate (when used) 0.3,
elitism = 1 (the best individual is copied automatically into the next population)

The stopping criterion was in all the cases the same: the number of function evaluations equal to 1000. Figure 6 shows a typical minimization history for Example 3 and BGA. In Figure 7 the average of these histories is shown.

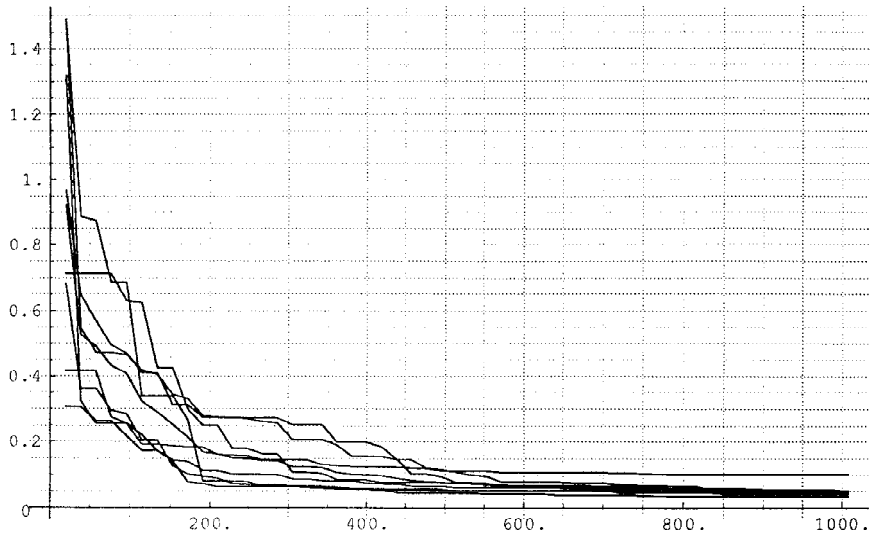

Figure 6.

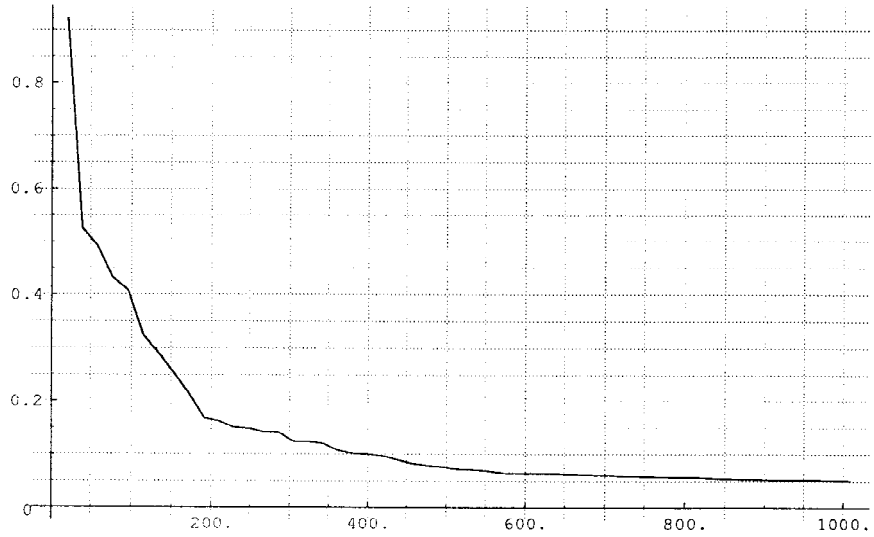

Figure 7.

In order to evaluate the convergence of the algorithms, three typical function values (levels) were chosen. For each algorithm and each function level the results are summarized in Tables 1, 2, 3.

The meaning of the columns is the following:

$J_i (i = 1, 2, 3)$ – the function level;

Table 1.

$-J_1$	MCRS			BGA			GA_{bu}		
	min	max	mean	min	max	mean	min	max	mean
88.20	480	730	560	291	–	465	349	–	755
88.25	500	830	680	320	–	552	465	–	–
88.30	680	–	920	320	–	784	465	–	–

$-J_1$	GA_{b2}			GA_{e2}			GA_{eu}		
	min	max	mean	min	max	mean	min	max	mean
88.20	153	–	590	204	–	–	59	–	–
88.25	153	–	590	262	–	–	59	–	–
88.30	153	–	–	–	–	–	59	–	–

Table 2.

$J_2 * (10^{-6})$	MCRS			BGA			GA_{bu}		
	min	max	mean	min	max	mean	min	max	mean
10	290	740	540	134	324	210	286	780	571
7	410	900	740	172	894	324	419	–	–
5	770	–	–	495	–	894	–	–	–

$J_2 * (10^{-6})$	GA_{b2}			GA_{e2}			GA_{eu}		
	min	max	mean	min	max	mean	min	max	mean
10	77	–	–	457	–	–	134	–	495
7	229	–	–	571	–	–	229	–	685
5	–	–	–	799	–	–	438	–	–

min(max) – minimal (maximal) number of function evaluations necessary to achieve the required level in all runs realized by the particular algorithm;

mean – the number of function evaluations necessary to achieve the required level of the average minimization history

If the level was not achieved in 1000 function evaluations, there is a dash in the corresponding position. Table 1, (2, 3) corresponds to Example 1, (2, 3), respectively.

The max column is interesting: if there are numeral characters there, then the required level was achieved in any run of the algorithm. On the contrary the min

Table 3.

J_3	MCRS			BGA			GA_{bu}		
	min	max	mean	min	max	mean	min	max	mean
0.4	140	–	920	20	172	115	37	964	325
0.2	770	–	–	115	362	191	118	–	–
0.1	–	–	–	172	–	419	640	–	–

J_3	GA_{b2}			GA_{e2}			GA_{eu}		
	min	max	mean	min	max	mean	min	max	mean
0.4	115	514	267	190	–	910	118	–	784
0.2	286	–	837	–	–	–	982	–	–
0.1	533	–	–	–	–	–	–	–	–

column in Table 1 documents that one run of GA_{eu} accidentally led to a better value then 88.3 after only 59 function evaluations.

Taking into account smoothness of the minimized functions, our examples can be listed as follows (starting from the most regular): Ex. 2, 1, 3. It is known that the MCRS gives better results when applied to smooth functions. This is seen from Tables 1 and 2 while from Table 3 we see that BGA is more successful than MCRS likely due to lack of smoothness of J_3. Looking at Tables 1, 2 we see that also here BGA worked very well using considerably fewer function evaluations than MCRS. From this point of view we may conclude that BGA was the most successful method for our class of optimization problems.

Acknowledgement

The authors acknowledge the support of the grant A1075707 of the Grant Agency of the Czech Academy of Sciences.

References

Brezzi, F. and Fortin, M. (1991), *Mixed and Hybrid Finite Element Methods*, Springer-Verlag, New York.

Daňková, J. and Haslinger, J. (1996), Numerical realization of a fictitious domain approach used in shape optimization. Part I: Distributed controls, *Appl. Math.* 41(2): 123–147.

Haslinger, J. and Jedelský, D. (1996), Genetic algorithms and fictitious domain based approaches in shape optimization, *Struct. Optimiz.* 12: 257–264.

Haslinger, J. and Klarbring, A. (1995), Fictitious domain/mixed finite element approach for a class of optimal shape design problems, *RAIRO* M^2AN 29(4): 435–450.

Haslinger, J., Maître, F. and Tomas, L. (1998), Fictitious domains methods with distributed Lagrange multipliers. Part I: Application to the solution of elliptic state problems. Part II: Application to the solution of shape optimization problems.

Haslinger, J. and Neittaanmäki, P. (1996), *Finite Element Approximation for Optimal Shape, Material and Topology Design*, Second Edition, J. Wiley & Sons, Chichester.

Pironneau, O. (1984), *Optimal Shape Design for Elliptic Systems*, Springer series in Computational Physics, Springer-Verlag, New York.

Tomas, L. (1997), *Optimisation de forme et domaines fictifs: Analyse de nouvelles formulations et aspects algorithmiques*, thesis, Ecole Centrale de Lyon.

Bäck, T. (1992), *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, New York.

Křivý, I. and Tvrdík, J. (1995), The Controlled Random Search Algorithm in Optimizing Regression Models, *Comput. Statist. and Data Anal.* 20: 229–234.

Křivý, I. and Tvrdík, J. (1996), *Stochastic Algorithms in Estimating Regression Models*, A. Prat (ed.), COMPSTAT 1996. Proceedings in Computational Statistics, Physica-Verlag 1996, Heidelberg, pp. 325–330.

Křivý, I. and Tvrdík, J. (1997), *Some Evolutionary Algorithms for Optimization*, MENDEL'97. 3rd International Mendel Conference on Genetic Algorithms, (Tech. Univ. of Brno, June 1997). Brno, PC-DIR 1997, pp. 65–70.

Nelder, J.A., Mead, R. (1964), A simplex method for function minimization, *Computer J.* 7: 308–313.

Price, W.L. (1976), A controlled random search procedure for global optimisation, *Computer J.* 20: 367–370.

Mühlenbein H., Schlierkamp-Voosen D. (1992), How Genetic Algorithms Really Work: Mutation and Hillclimbing, In Männer, R. & Manderick, B. (eds.), *Parallel Problem Solving From Nature*, North-Holland, Amsterdam, pp. 15–26.

Mühlenbein H., Schlierkamp-Voosen D. (1993), Predictive models for the breeder genetic algorithm, I. Continuous parameter optimization, *Evolutionary Computation* Vol. 1, No. 1, pp. 25–49.