A New Method of Centers Location in Gaussian RBF Interpolation Networks

Marek Bazan and Ewa Skubalska-Rafajłowicz

Institute of Computer Engineering, Automatics and Robotics,
Department of Electronics, Wrocław University of Technology, Poland
{marek.bazan,ewa.rafajlowicz}@pwr.wroc.pl

Abstract. In this paper we present a new method of obtaining near-optimal points sets for interpolation by Gaussian radial basis functions networks. The method is based on minimizing the maximal value of the power function. The power function provides an upper bound on the local RBF interpolation error. We use Latin hypercube designs and a space-filling curve based space-filling designs as starting points for the optimization procedure. We restrict our attention to 1-D and 2-D interpolation problems. Finally, we provide results of several numerical experiments. We compare the performance of this new method with the method of [6].

Keywords: radial bases function network, interpolation, error bound, power function, computer experiment design.

1 Introduction

Radial bases functions (RBF) were introduced in the solution of the multivariate interpolation problem (see [26] and references cited therein). Radial basis function networks are two-layer feed-forward networks with RBFs as activation functions in the hidden units, and linear activation functions in the output units [22], [4], [25]. It is well known that Gaussian RBF networks can approximate any continuous mapping on a compact domain [23]. RBF networks have been extensively applied to pattern recognition [3], function approximation [23], [18], probability density function estimation [3], [37], [38], regression function estimation [25], [14], [19], approximating the boundary of an object in a binary image [29], to speed up deterministic search algorithms used for the local optimization [1], [2], to global optimization in connection with local deterministic procedures [15], [30] and creating surfaces using radial basis functions from scattered data [8], [20], among many others.

The purpose of interpolating RBF networks is to approximate functions $f : \mathcal{R}^d \to \mathcal{R}$ that are given as data $\{f(x_i)\}_{i=1:N}$ on a finite set $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \subset \Omega \subset \mathcal{R}^d$ of distinct points (centers, nodes, knots) by expression

$$s(\mathbf{x}) = \sum_{i=1}^{N} w_i \varphi_\beta(||\mathbf{x} - \mathbf{x}_i||), \tag{1}$$

L. Rutkowski et al. (Eds.): ICAISC 2013, Part I, LNAI 7894, pp. 20–31, 2013.

where $||.||$ denotes the Euclidean distance between two points in $\mathcal{R}$ and $\varphi_\beta(r) = \exp(-\beta r^2)$ is a one dimensional Gaussian function defined for a shape parameter $\beta = \frac{1}{2\sigma^2} > 0$. In the interpolant (1) we simply compute the coefficients w_i as the solution of the linear system

$$A_{\mathbf{X}}\mathbf{w} = \mathbf{f} \tag{2}$$

where $A_{\mathbf{X}} = (\varphi_\beta(||\mathbf{x}_i - \mathbf{x}_j||))_{i,j=1}^N$ is the information matrix for the set of nodes $\mathbf{X}$, $\mathbf{w} = (w_1, \ldots, w_N)^T$ and $\mathbf{f} = (f_1, \ldots, f_N)^T$.

Gaussian radial bases functions are infinitely smooth (C^∞) and analytic functions. Moreover the Gaussian interpolation matrix $A_{\mathbf{X}}$ is positive definite (thus it is invertible) if the centers are distinct [21]. If $A_{\mathbf{X}}$ is positive definite for any $\mathbf{X} \subset \Omega$ (consisting of distinct points), then φ_β is said to be positive definite.

Franke [10] found that it is very sensitive to the choice of parameter β. It is known that the Gaussian radial functions are susceptible to Runge's phenomenon, however there exist interpolation node distributions that prevent such oscillatory behavior of the solution and allow stable interpolation [24]. Usually interpolation matrix $A_{\mathbf{X}}$ is very ill-conditioned, and the weights obtained by solving (2) yield an interpolation mapping $s(x)$ that exhibits oscillatory behavior in between data points. Furthermore, the conditioning of the interpolation matrix grows with the problem size N, since the condition number of interpolation matrix $A_{\mathbf{X}}$, defined as $||A_{\mathbf{X}}||_2||A_{\mathbf{X}}^{-1}||_2$ (where $||.||_2$ is a spectral metric), depends mostly on $min_{x_i,x_j \in \mathbf{X}}||x_i - x_j||$. If we fix the number of nodes N the only factor which is important in the balance between the accuracy of interpolation and the conditioning of numerical computations is the shape parameter β. The dependence of the condition number on β parameter is less crucial, however too small values of β (too large values of σ) may result in instability of interpolation due to a bad conditioning. Moreover, it is possible to use the preconditioning methods which allows for the stable computation of Gaussian radial basis function interpolants (see for example [12], [9]).

It is known that, in general, the attainable error and the condition of the interpolation matrices cannot both be kept small [34]. However, this property is based on upper bounds of both factors only.

In the interpolation problems a proper choice of interpolation nodes, i.e., the proper experiment design, is essential for good approximations. It is advisable to keep the number of interpolation nodes at a reasonable level (N should be not too large), because it allows the controlling of the condition number of the interpolation matrix. Nevertheless, in many cases, the more important factor is the cost of the function evaluation. For example, in deterministic computer simulations, which are becoming widely used in science and engineering, the simulation model is often replaced by an approximating model, based on simulations in some points. Thus, the problem of node placement design for RBF interpolation should be considered also in the general context of experiment and computer experiment design methods. It is known that such type designs should at least be space-filling in some sense. With no additional assumptions, it is important to obtain information from the entire design space. Therefore, design points should be 'evenly spread' over the entire region. Several space-filling

criteria are discussed in the literature [33], [17]. The design is often restricted to a d-dimensional grid of n levels in every dimension, i.e., such that for each dimension j all nodes coordinate x_{ij} are distinct. Such a design is called a Latin hypercube design (LHD) [33], [30]. The generation of so called low discrepancy sequences is also often used for space-filling points generation. Faure, Halton and Sobol sequences, are increasingly popular in computer experiments [33], [27], [39]. A maximin space-filling design is a set of points such that the separation distance (i.e. the minimal distance among pairs of points) is maximal. Notice, that the maximization of the separation distance influences positively the condition number of the design based RBF interpolation.

In this paper we propose a new method of obtaining near-optimal points sets for interpolation by Gaussian radial basis functions networks. Motivated by methods of selecting RBF centers based on placing prototypes of the centers initially at equidistributed (EQD) points generated along Sierpiński and Hilbert space-filling curves [28], [19], [39], [36], we propose a minmax optimization procedure which uses these space-filling curve based space-filling designs as starting points for minimizing the maximal value of the power function introduced by Schaback [41], [34], [6], [7].

We restrict our attention to the interpolation problems on a cube in $\mathcal{R}^d$, i.e., we assume that $\Omega = [-1, 1]^d$.

The paper is organized as follows. Section 2 provides known local error estimates for interpolation by radial basis functions. Section 3 formulates the minmax optimization problem leading to optimal center location. In section 4 we present algorithms used for the construction of near-optimal set of interpolation centers. Section 5 is devoted to numerical tests. Seven out of eight designs obtained using the proposed new method give upper bound values lower than that generated using the greedy algorithm presented in [6].

2 Interpolation Error Bounds

A general error bound for the interpolation of function $f : \mathbb{R}^d \to \mathbb{R}$ with (1) is derived (c.f. [41], [34], [40], page 176) using the Lagrange (cardinal) basis for the interpolation function, i.e.:

$$s(\mathbf{x}) = \sum_{i=1}^{N} u_i(\mathbf{x}) f_i,$$

where u_i is a certain continuous function $u_i : \Omega \to \mathcal{R},\ i = 1, \ldots, N$ such that:

$$u_i(\mathbf{x}_j) = \delta_{ji}, \quad \text{i.e.} \qquad u_i(\mathbf{x}) = \begin{cases} 1 \text{ for } \mathbf{x} = \mathbf{x}_i \\ 0 \text{ for } \mathbf{x} = \mathbf{x}_j \quad \text{and} \quad j = 1 \ldots N \quad \text{and} \quad j \neq i. \end{cases} \tag{3}$$

The existence of a vector $\mathbf{u}(x) = (u_1(\mathbf{x}), \ldots, u_N(\mathbf{x}))^T$ for a given $\mathbf{x}$ was shown in [41]. Notice that u depends on φ_β. Let

$$R(\mathbf{x}) := (\varphi_\beta(||\mathbf{x} - \mathbf{x}_1||), \varphi_\beta(||\mathbf{x} - \mathbf{x}_2||), \ldots, \varphi_\beta(||\mathbf{x} - \mathbf{x}_N||))^T.$$

Then the vector $\mathbf{u}(\mathbf{x})$ is a solution of the system of equations

$$A_{\mathbf{X}}\mathbf{u}(\mathbf{x}) = R(\mathbf{x}). \tag{4}$$

The interpolation error $|f(x) - s(x)|$ can be bounded by

$$|f(x) - s(x)| \leq P_{\varphi_\beta,\mathbf{X}}(x)|f|_{\mathcal{N}_{\varphi_\beta}(\Omega)},$$

where $P_{\varphi_\beta,\mathbf{X}}$ is a power function defined as

$$P_{\varphi_\beta,\mathbf{X}}(\mathbf{x})^2 := \varphi_\beta(0) - 2\sum_{j=1}^{N} u_j(\mathbf{x})\varphi_\beta(||\mathbf{x} - \mathbf{x}_j||) + \sum_{i,j=1}^{N} u_i(\mathbf{x})u_j(\mathbf{x})\varphi_\beta(||\mathbf{x}_i - \mathbf{x}_j||) \tag{5}$$

and $|\cdot|_{\mathcal{N}_{\varphi_\beta}(\Omega)}$ is a norm in the native space generated by the radial basis function φ_β (reproducing kernel Hilbert space with φ_β as its reproducing kernel) [35]. Notice, that one can calculate a value of the power function (5) for any point of the domain Ω using (4). For a given radial basis function the power function (5) depends only on a location of data points from $\mathbf{X}$ within the domain Ω.

The interpolation matrix $A_{\mathbf{X}}$ is ill-conditioned and therefore to solve the system (4) we use the singular value decomposition of the matrix $A_{\mathbf{X}}$ (c.f. [13]). Solving (4) for any $\mathbf{x} \in \Omega$ one can calculate a value of the power function (5) for any point of the domain Ω. Checking whether the obtained $\mathbf{u}(\mathbf{x})$ has the property (3) enables us to choose the shape parameter β for the radial function φ_β. Appropriate choice of the value of the shape parameter is a different method of improving the conditioning of the matrix $A_{\mathbf{X}}$ than proposed in [9].

3 A Min-max Problem Formulation

From (2) one can conclude that $\max_{x\in\Omega} P_{\varphi_\beta,\mathbf{X}}(\mathbf{x})$ (Ω is a compact set) is the most important factor in the upperbounding of the interpolation error in the supremum norm.

Let

$$\underline{\mathbf{X}} = \left(x_1^{(1)}, x_1^{(2)}, \ldots, x_1^{(d)}, \ldots, x_N^{(1)}, x_N^{(2)}, \ldots, x_N^{(d)}\right) \in \Omega^N \subset \mathcal{R}^{N\cdot d}$$

represents an interpolation design $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ where $\mathbf{x}_i = (x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(d)})$ $(i = 1, \ldots, N)$. Notice, that in fact, $\underline{\mathbf{X}}$ is a concatenated vector of all design points contained in the set $\mathbf{X}$.

Let us define function $G : \mathbb{R}^{N\cdot d} \times \mathbb{R}^d \to \mathbb{R}$ as

$$G(\underline{\mathbf{X}}; \mathbf{x}) = P_{\varphi_\beta,\mathbf{X}}(\mathbf{x})^2. \tag{6}$$

An optimal design (location of centers) for the radial basis interpolation network with the radial basis function φ_β in the domain Ω can be defined as a global minimum with respect to $\max_{\mathbf{x}\in\Omega} P_{\varphi_\beta,\mathbf{X}}(\mathbf{x})^2$. Let us denote $F(\underline{\mathbf{X}}) = ||G(\underline{\mathbf{X}}; \cdot)||_{sup}$. Then the minimization problem to find an optimal design reads:

$$\min_{\underline{\mathbf{X}} \in \Omega^N} F(\underline{\mathbf{X}}) = \min_{\underline{\mathbf{X}} \in \Omega^N} ||G(\underline{\mathbf{X}}; \cdot)||_{sup} = \min_{\underline{\mathbf{X}} \in \Omega^N} [\max_{\mathbf{x} \in \Omega} G(\underline{\mathbf{X}}; \mathbf{x})]. \quad (7)$$

4 Algorithms to Solve the Max and the Min Problems

Solving (7) consists of two optimization algorithms. An inner algorithm is used to find the maximum value of the squared power function $P_{\varphi_\beta, \mathbf{X}}(\mathbf{x})^2$ (6) on Ω for a given design $\mathbf{X}$. An outer algorithm is used to find a concatenated vector representing a design $\mathbf{X}$ which minimizes $F(\underline{\mathbf{X}})$ on the domain Ω^N, i.e., it produces an optimal (or at least near-optimal) interpolation design consisting of exactly N centers.

The outer optimization problem is a constrained minimization problem. The values of $\max_{\mathbf{x} \in \Omega} G(\underline{\mathbf{X}}_k; \mathbf{x})$ are obtained as a result of a simple scan of the domain Ω. Following the approach presented in [6], we maximize over some very large discrete set $Y \subset \Omega$ instead of maximizing on Ω.

To minimize $\min_{\underline{\mathbf{X}} \in \Omega^N} F(\underline{\mathbf{X}})$ we have used a non-gradient algorithm, as not much is known about properties of F.

4.1 A Constrained Non-linear Programming Algorithm

Every configuration $\underline{\mathbf{X}}$ in which at least one design point $\mathbf{x}_i$ $(i = 1, \ldots, N)$ lies on the boundary $\delta\Omega$ of the domain Ω is on the constraint boundary of the minimization problem (7). The most common approach to deal with constraint violation is to add a penalty term that depends on how much the constraint is violated. If a current design contains points that are outside the domain Ω we have to project these points onto the boundary $\delta\Omega$.

In numerical experiments presented in this paper $\Omega = [-1, 1] \times [-1, 1]$. We do not use the most common projection method which is the orthogonal projection defined as $\underline{\mathbf{X}}^{(proj)} = \arg\min_{\underline{\mathbf{Y}} \in \delta\Omega} ||\underline{\mathbf{X}} - \underline{\mathbf{Y}}||$. Such a projection transforms points from $\underline{\mathbf{X}}$ that lie outside Ω for which two constraints are violated into the closest corner of the domain. Such a situation automatically generates a singular design matrix if there are at least two points that are transformed into the same corner. It is not a rare situation. The projection for which the overlapping of transformed centers is less probable is to choose a point on the boundary which lies on the interval between the point $\mathbf{x}_i$ to be transformed and the center of the domain Ω.

Now, we can calculate (6) for $\underline{\mathbf{X}}^{(proj)}$ and add to $F(\underline{\mathbf{X}}^{(proj)})$ the penalty proportional to $||\underline{\mathbf{X}}^{(proj)} - \underline{\mathbf{X}}||$. Nevertheless, for the sake of simplicity, we omit a penalty term using only projection. This is motivated by the fact that adding a penalty term creates an artificial piece of information which is useless from the optimization point of view. The power function defined outside the domain Ω provides upperbounds for an interpolation problem defined on other domains than Ω.

In numerical experiments we have achieved the best values of the objective function F using the Rosenbrock algorithm initially described in [31] and modified by Jacob [16].

4.2 A Stable Calculation of the Objective Function in the Inner Algorithm

The calculation of (6) which is the objective function for the inner algorithm requires a stable calculation of the Lagrange representation $\mathbf{u}(\mathbf{x})$ for any $\mathbf{x} \in \Omega$ by solving (4). For $N > 40$ and for small values of shape parameter β (too large values of σ, i.e., close to a diameter of the domain space Ω) in Gaussian basis it may happen that a calculated solution $\mathbf{u}(\mathbf{x})$ is not a cardinal solution (see property (3)).

Problems with a stable solution of (4) for the Gaussian with small values of a shape parameter were reported in e.g. [9]. Here we propose to choose the largest possible value of parameter δ that enables us to find a solution $\mathbf{u}(\mathbf{x})$ which guaranties that the property (3) is maintained at the level of the absolute error of 10^{-7}. In the example presented in the next section (for $N = 54$) σ value set to the quarter of the diameter of $\underline{\mathbf{X}}_k$ is sufficient.

4.3 Initial Designs

As mentioned in the previous paragraph we limit our numerical experiments to $\Omega = [-1, 1]^2$. To generate $\underline{\mathbf{X}}_0$ we use four different methods:

1. Space-filling latin hypercube sampling method [33], [30].
2. A uniform design based on the Sierpiński space-filling curve [36].
3. A uniform design based on the Hilbert space-filling curve [36].
4. Quasi-optimal design obtained by the greedy algorithm due to De Marchi, Schaback and Wendland (DMSW) [6].

The advantage of initial configurations generated with the first three methods is that they are the initial configuration of N points in $\Omega \subset \mathbb{R}^d$ and N does not need to be a d-th power of a natural number. The fourth initial distribution is used in this paper not only as a starting point for the optimization but also as a reference for comparisons. Up to our knowledge the designs $\underline{\mathbf{X}}_0$ generated with the algorithm DMSW produce the lowest values of $||P_{\varphi_\beta,\mathbf{X}}(\mathbf{x})^2||_{sup}$ over Ω among results published in the literature so far for the methods based on minimizing the power function (5).

5 Numerical Results

5.1 One Dimensional Example

In order to show that the proposed optimization algorithm produces designs that give a lower interpolation error we interpolate function $y = (1+(5x)^2)^{-1}$ using a

Gaussian RBF network with 9 centers on the interval $[-1, 1]$. Radius parameter σ of Gaussians was set to 1. Figure 1 shows the interpolation curves obtained for three configurations of centers:

1. obtained by DMSW method [6],
2. centers are scaled zeros of the 9-th Tshebyshev polynomial of the first kind,
3. obtained by our optimization algorithm.

The smallest maximal error is equal to 0.16098 and is obtained by using the method proposed in the paper. The resulting error is slightly smaller than the maximal error for the Tshebyshev centers which is equal to 0.16489. The results obtained by DMSW algorithm [6] are worse with the error supremum equals to 0.21884. The new configuration decreases the Runge phenomenon compared to the configuration obtained by DMSW algorithm [6].

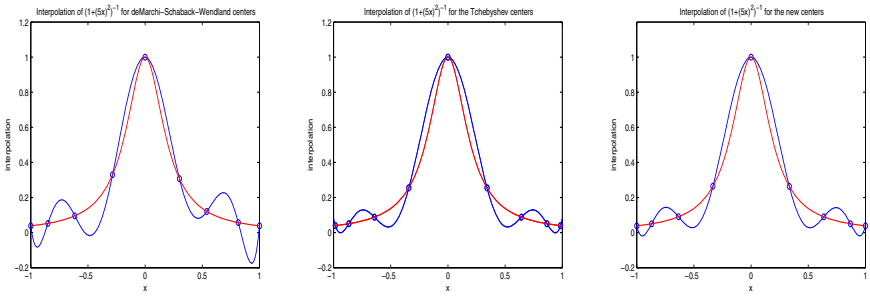

Fig. 1. Interpolation of the function $y = (1 + (5x)^2)^{-1}$ with the Gaussain radial basis function with a shape parameter equals to 1 on 9 nodes from the interval $[-1, 1]$ on three different node configurations a) obtained by DMSW algorithm [6] b) Tchebyshev knots c) obtained by our optimization algorithm

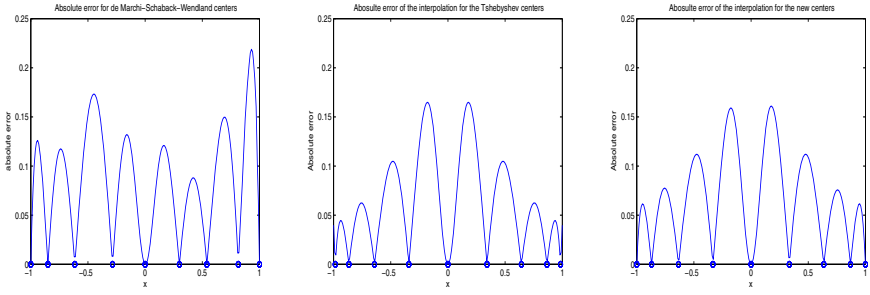

Fig. 2. Interpolation error for the interpolation of the function $y = (1 + (5x)^2)^{-1}$ with the Gaussain radial basis function with a shape parameter equals to 1 on 9 nodes from the interval $[-1, 1]$ on three different node configurations a) obtained by DMSW algorithm [6] b) Tchebyshev knots c) obtained by our optimization algorithm

5.2 Two Dimensional Example

In this section we present the performance of the proposed design optimization algorithm launched from 5 randomly generated points forming a Latin hypercube [33], one optimization started from a Sierpiński filling curve configuration of centers [36], another one started from a Hilbert filling curve configuration of centers [36] and one optimization started from the quasi-optimal configuration of centers generated by DMSW algorithm [6].

To guarantee that the objective function F is not too small and the nonlinear optimisation algorithm runs properly we use a scaling factor of 10^7 in the function calculations, i.e., instead of minimizing F we minimize $10^7 F$.

Table 1 shows the results obtained for eight different starting configurations: five initial designs were generated using the Latin hypercube method, one initial design was a uniform design along the Sierpiński space-filling curve [36], the next initiali design was s uniform design along the Hilbert filling space curve [36] and the last one was generated by DMSW quasi-optimal method [6]. As one can see from this table in four out five different Latin hypercube starting configurations the proposed scheme was able to decrease the objective function value below the value calculated for the configuration at the starting configuration generated using the quasi-optimal method. Also starting from the Sierpiński and the Hilbert configuration the result was better although it was not better than for the successful Latin hypercube configurations.

Table 1. Results of numerical experiments for 54 knots placed in $\Omega = [-1, 1] \times [-1, 1]$. The first five rows were obtained using the proposed algorithm started from configurations generated using the Latin hypercube method [33]. The sixth row presents results for the Sierpiński space-filling curve based starting configuration [36]. The seventh row presents results for the Hilbert space-filling curve based starting configuration [36]. In the eighth row the optimization was started from the quasi-optimal configuration generated by DMSW method [6]. The first column describes the way the starting configuration was generated, the second one shows the objective function value F at the starting configuration and the third one shows the number of the objective function calculations to achieve the value that is shown in the fourth column.

starting configuration generation method	starting configuration objective f. val.	num. of obj. f. calculations	final configuration objective f. val.
Latin hypercube 1	8693.840221	23778	8.713339
Latin hypercube 2	24004.490524	63693	9.163693
Latin hypercube 3	21275.298287	34164	7.821087
Latin hypercube 4	63583.290885	37070	9.446265
Latin hypercube 5	22191.814212	37195	23.794642
Sierpiński filling curve	15117.265424	42476	13.522486
Hilbert filling curve	24429.318636	36652	12.395903
quasi-optimal	20.815944	36679	10.736679

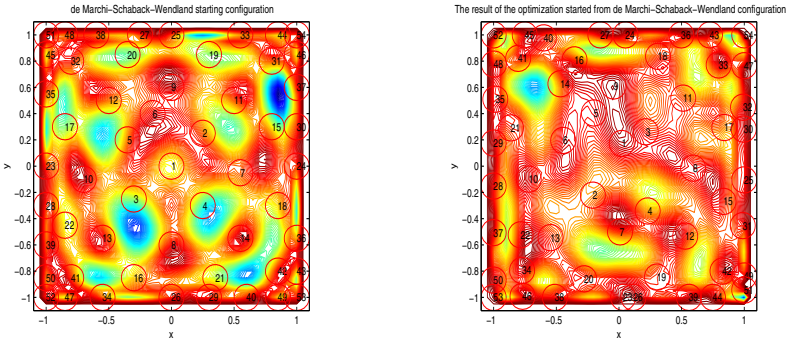

Fig. 3. Left) The quasi optimal configuration of 54 centers generated using DMSW method [6]. Right) The configuration obtained by our algorithm started from the quasi optimal configuration. Both configurations are plotted together with isolines of the squared power function.

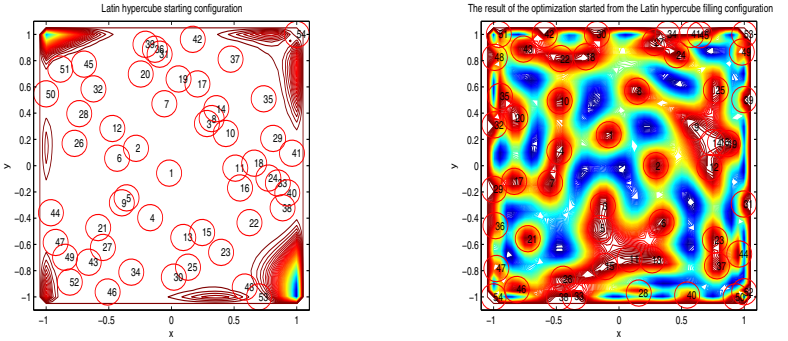

Fig. 4. Left) An example of latin hypercube configuration of 54 nodes generated using the algorithm from [33]. Right) The configuration obtained using by our optimization algorithm started from the the latin hypercube based configuration. Both configurations are plotted together with isolines of the squared power function.

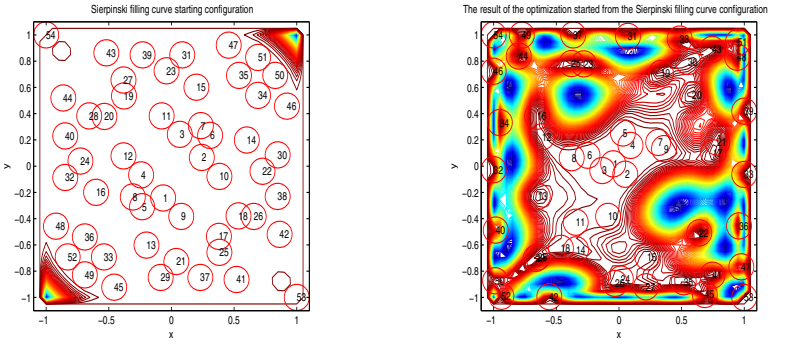

Fig. 5. Left) An example of a Sierpinski filling curve configuration of 54 nodes (c.f. [36]). Right) The design obtained by our optimization algorithm started from the Sierpiński space-filling curve based configuration. Both designs are plotted together with isolines of the squared power function.

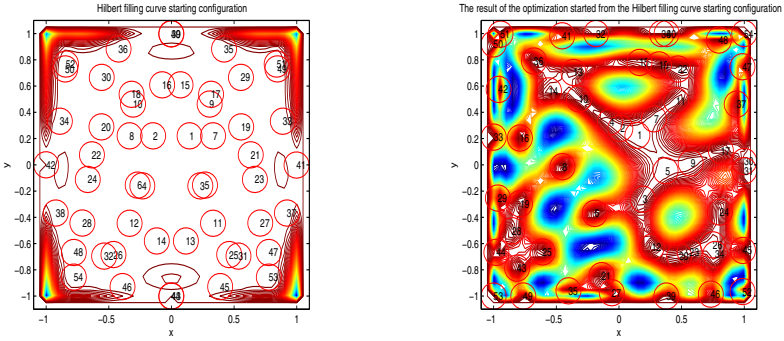

Fig. 6. Left) An example of a Hilbert filling curve configuration of 54 nodes (c.f. [36]). Right) The design obtained by our optimization algorithm started from the Hilbert space-filling curve based configuration. Both designs are plotted together with isolines of the squared power function.

6 Conclusions

We have presented a scheme of using constrained non-linear programming to calculate the optimal configurations of centers for a radial basis interpolation process. From eight designs obtained by the proposed optimization algorithm seven configurations give objective function values lower than that generated as an initial configuration by DMSW algorithm [6].

The work in this paper focused on the Gaussian kernel. There are many other positive definite kernels: inverse multiquadric [21], Wendland function [40], Sobolev splines [34] and others that can be constructed using a method described in [35]. It should be mentioned that the method proposed in this paper can be applied to other strictly positive definite radial bases function systems. Furthermore, the proposed method can be used also when we have to extend the near-optimal design by a new additional point.

Acknowledgment. Calculations have been carried out in the Wrocław Centre for Networking and Supercomputing (`http://www.wcss.wroc.pl`) grant No 205, using MATLAB 2012.

References

1. Bazan, M., Russenschuck, S.: Using neural networks to speed up optimization algorithms. Eur. Phys. J., AP 12, 109–115 (2000)
2. Bazan, M., Aleksa, M., Russenschuck, S.: An improved method using radial basis function neural networks to speed up optimization algorithms. IEEE Trans. on Magnetics 38, 1081–1084 (2002)
3. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer Science+Business Media, New York (2006)
4. Broomhead, D.S., Lowe, D.: Multivariable Functional Interpolation and Adaptive Networks. Complex Systems 2, 321–355 (1988)

5. van Dam, E.: Two-dimensional minimax Latin hypercube designs. Discrete Applied Math. 156(18), 3483–3493 (2007)
6. De Marchi, S., Schaback, R., Wendland, H.: Near-optimal data-independent points locations for radial basis function interpolation. Adv. in Comp. Math. 23(3), 317–330 (2003)
7. De Marchi, S.: On optimal center locations for radial basis function interpolation: computational aspects. Rend. Sem. Mat. Univ. Pol. Torino, Splines and Radial Functions 61(3), 343–358 (2003)
8. Dyn, N., Levin, D., Rippa, S.: Numerical Procedures for Surface Fitting of Scattered Data by Radial Basis Functions. SIAM Journal of Scientific and Statistical Computing 7(2), 639–659 (1986)
9. Fasshauer, G.E., Zhang, J.G.: Preconditioning of Radial basis function interpolation systems via accelerated iterated approximate moving least squares approximation. In: Ferreira, A.J.M., et al. (eds.) Progress in Meshless Methods. Springer (2009)
10. Franke, R.: Scattered Data Interpolation, Test of Some Methods. Mathematics of Computation 38, 181–200 (1982)
11. Fornberg, B., Wright, G., Larsson, E.: Some observations regarding interpolants in the limit of flat radial basis functions. Computers Math. Applic. 47(1), 37–55 (2004)
12. Fornberg, B., Larsson, E., Flyer, N.: Stable computations with Gaussian radial basis functions. SIAM J. Sci. Comput. 33(2), 869–892 (2011)
13. Hansen, P.C.: Rank-deficient and discrete ill-posed problems. SIAM, Philadelphia (1998)
14. Girosi, F., Jones, M., Poggio, T.: Regularization theory and neural networks architectures. Neural Computation 7(2), 219–269 (1995)
15. Ishikawa, T.T., Matsunami, M.: A combined method for global optimization using radial basis function and deterministic approach. IEEE Trans. Magn. 35, 1730–1733 (1999)
16. Jacob, H.G.: Rechnergestützte Optimierung statischer und dynamischer Systeme. Springer (1982)
17. Kleijnen, J.P.C.: Design and Analysis of Simulation Experiments. Springer (2009)
18. Kainen, P.C., Kurková, V., Sanguineti, M.: Complexity of Gaussian-radial-basis networks approximating smooth functions. Journal of Complexity 25(1), 63–74 (2009)
19. Krzyżak, A., Skubalska-Rafajłowicz, E.: Combining space-filling curves and radial basis function networks. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI), vol. 3070, pp. 229–234. Springer, Heidelberg (2004)
20. Lin, G.F., Chen, L.H.: A spatial interpolation method based on radial basis function networks incorporating a semivariogram model. Journal of Hydrology 288, 288–298 (2004)
21. Micchelli, C.A.: Interpolation of Scattered Data: Distance Matrices and Conditionally Positive Definite Functions. Constructive Approximation 2, 11–22 (1986)
22. Moody, J., Darken, J.: Fast learning in networks of locally-tuned processing units. Neural Computation 1, 281–294 (1989)
23. Park, J., Sandberg, I.W.: Approximation and radial-basis-function networks. Neural Computation 5(2), 305–316 (1993)
24. Platte, R.B., Driscoll, T.A.: Polynomials and potential theory for Gaussian radial basis function interpolation. SIAM J. Numer. Anal. 43(2), 750–766 (2005)

25. Poggio, T., Girosi, F.: A Theory of Networks for Approximation and Learning, AI Lab, MIT, AI Memo 1140 (1989)
26. Powell, M.J.D.: Radial basis functions for multivariable interpolation: a review. In: Algorithms for Approximation. Claredon Press, Oxford (1987)
27. Rafajłowicz, E., Schwabe, R.: Halton and Hammersley sequences in multivariate nonparametric regression. Statistics and Probability Letters 76, 803–812 (2006)
28. Rafajłowicz, E., Skubalska-Rafajłowicz, E.: RBF nets based on equidistributed points. In: Proc. of 9th IEEE International Conf. Methods and Models in Automation and Robotics MMAR 2003, vol. 2, pp. 921–926 (2003)
29. Rafajłowicz, E., Skubalska-Rafajłowicz, E.: RBF nets for approximating an object's boundary by image random sampling. Nonlinear Analysis: Theory, Methods and Applications 71(12), 1247–1254 (2009)
30. Regis, R.G., Shoemaker, C.A.: Improved Strategies for Radial basis Function Methods for Global Optimization. J. of Global Optimization 37(1), 113–135 (2007)
31. Rosenbrock, H.H.: An automatic method for finding the greatest or least value of function. The Computer Journal 3(3), 175–184 (1960)
32. Sacks, J., Welch, W.J., Mitchell, T.J., Wynn, H.P.: Design and Analysis of Computer Experiments. Statistical Science 4(4), 409–435 (1989)
33. Santner, T.J., Williams, B.J., Notz, W.I.: The Design and Analysis of Computer Experiments. Springer Series in Stastistics (2003)
34. Schaback, R.: Error estimates and condition numbers for radial basis function interpolation. Advances in Computational Mathematics 3, 251–264 (1995)
35. Schaback, R.: Native Hilbert spaces for radial basis functions I. In: Müller, M.W., et al. (eds.) New Developments in Approximation Theory, 2nd Int. Dortmund Meeting (IDoMAT) 1998. Int. Ser. Numer. Math., vol. 132, pp. 255–282. Birkhäuser Verlag, Basel (1999)
36. Skubalska-Rafajłowicz, E.: Space-filling Curves in Multivariate Decision Problems. Wrocław University of Technology Press, Wrocław (2001) (in Polish)
37. Skubalska-Rafajłowicz, E.: RBF Neural Network for Probability Density Function Estimation and Detecting Changes in Multivariate Processes. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 133–141. Springer, Heidelberg (2006)
38. Skubalska-Rafajłowicz, E.: Random Projection RBF Nets for Multidimensional Density Estimation. Int. J. Appl. Math. Comput. Sci. 18(4), 455–464 (2008)
39. Skubalska-Rafajłowicz, E., Rafajłowicz, E.: Sampling multidimensional signals by a new class of quasi-random sequences. Multidimensional System and Signal Processing 23, 237–253 (2012)
40. Wendland, H.: Scattered Data Approximation. Cambridge University Press (2005)
41. Wu, Z., Schaback, R.: Local error estimates for radial basis function interpolation of scattered data. IMA J. Numer. Anal. 13, 13–27 (1993)