APPLICATION OF REDUCED-BASIS METHODOLOGIES IN PARTICLE SWARM OPTIMIZATION ALGORITHM FOR TRUSS DESIGN

Leonardo Correia de Oliveira, leoc.deoliveira@gmail.com
Silvana Maria Bastos Afonso, smb@ufpe.br
Universidade Federal de Pernambuco, Avenida Acadêmico Hélio Ramos, S/N, Recife-PE-50740-530

Samuel Arimateia de Souza, samuelbighead@gmail.com
Universidade Federal de Pernambuco, Avenida Acadêmico Hélio Ramos, S/N, Recife-PE-50740-530

Abstract. *Gradient based algorithms are in general very efficient in locating a relative optimum closest to the starting point of the design space. However, in recent years non gradient probabilistic based algorithms have been extensively investigated by the research community. Such algorithms are search procedures based on some natural phenomena. However such methodologies are involved with many more function evaluations than gradient based strategies. Despite of that, they are very versatile as they are easy to implement, accept both discrete and continuous design variables, do not require function continuality and above of all are very suitable for parallel computation. The recent probabilistic algorithm called Particle Swarm Optimizer is implemented in this work. As applications of optimization tools to large structural design problems could be cost prohibitive unless suitable approximation concepts are employed, problem approximation using the reduced-basis method (RBM) is here considered. The RBM is a Galerkin projection onto low order approximation spaces comprising solutions of the problem of interest at selected points in the parameter/design space. Moreover, the accuracy of such quantities is assured as errors estimators are incorporated in the procedure. The particle swarm optimization algorithm is then developed in the framework of the RBM. Truss structuress are the applications addressed here.*

Keywords*: optimization, particle swarm, reduced basis*

1. INTRODUCTION

Optimization techniques have been extensively used to obtain economical and reliable structural designs. Commonly in an optimization process several functions evaluations are required in consequence of design changes dictated by optimizers. To apply such techniques to real engineering problems could be an issue as the computational cost required for multiple numerical simulations could be, in certain cases, prohibitive. In the last decade, approximation concepts (Haftka et al, 2004) started to be more extensively applied in optimization procedures as a strategy to overcome such drawback. The approximations methodologies can be divided into two types: functional and physical (Haftka et al, 2004). While in the first an alternate and explicit expression is sought for the objective function and/or the constraints of the problem, the focus of the second is on replacing the original problem by one which is approximately equivalent but which is easier to be solve.

Problem approximation is the focus of the present work and, in this context, we will consider the reduced-basis (RB) method (Almroth et al, 1978, Noor and Peters, 1980, Fink and Rheinboldt, 1983, Prud'homme et al, 2002). The purpose of such scheme is to get high fidelity model information with low computational cost. The central idea considered in the development of the method is the recognition that to approximate the field variable and hence the outputs of the solution we do not need to represent every possible function of the infinite dimensional solution space associated with the governing equation, but rather those on a much lower-dimensional manifold induced by the parametric dependence.

To obtain structural designs a computational system using a standard SSO algorithm (Afonso, 1995) is implemented integrating geometric definition, finite element (FE) analysis and optimization. In the present context, RB approximations rule the structural analysis module. As a consequence, output is very fast computable. Single objective optimization problems will be addressed here. The particle swarm (PS) algorithm is used as an optimizer.

A fully FE based SSO algorithm is also implemented for comparison purposes. Some examples are solved and the results are discussed. The importance of considering RB approximations for functions evaluations in an optimization procedure is highlighted.

2. PROBLEM FORMULATION

The typical formulation of a structural optimization problem is given by:

$$\text{Minimize or maximize } f(\mathbf{x})$$

$$\text{subjected to } \begin{cases} g_i(\mathbf{x}) \le 0 & i = 1 \ldots n_{con} \\ x_k^l \le x_k \le x_k^{up} & k = 1 \ldots n_{var} \end{cases} \tag{1}$$

where $f(\mathbf{x})$ is the objective function, $\mathbf{x}$ is the vector of design variables, x^l and x^{up} are the limit values for the design variable n_{var} is the number of design variables, g is a typical constraint and n_{con} is the number of constraints.

3. PARTICLE SWARM OPTIMIZATION

3.1. General idea

PS refers to a relatively new family of algorithms that may be used to find optimal (or near optimal) solutions to numerical and qualitative problems (Pomeroy, 2003). PSO was originally developed by a social-psychologist (James Kennedy) and an electrical engineer (Russell Eberhart) in 1995.

The Particle Swarm Optimization (PSO) (Sobieszczanski-Sobieski and Venter, 2002) is based on natural phenomena, which is a simplified social model that is closely tied to swarming theory (Pomeroy, 2003). A physical analogy might be a swarm of bees that is adapting to its environment. In this analogy each bee (a particle) makes use of its own memory as well a knowledge gained by the swarm as a whole, to efficiently adapt to its environment.

The numerical implementation of a social model assumes that population is a swarm and each individual is a particle. The algorithm repeatedly updates the position of each particle over a time period to simulate the adaptation of the swarm to the environment. In order to find a new position of a particle, its current position, a velocity vector and a time increment are taken into account (Bochnek and Forys, 2003).

3.2. Basic algorithm

Particle swarm optimization makes use of a velocity vector to update the current position of each particle in the swarm. The position of each particle is updated based on the social behaviour that a population of individuals, the swarm, adapts to its environment by returning to promising regions that were previously discovered. The process is stochastic in nature and makes use of the memory of each particle as well as the knowledge gained by the swarm as a whole. The outline of a basic PSO algorithm is as follows:

1- Start with an initial set of particles, typically randomly distributed throughout the design space;
2- Calculate a velocity vector for each particle in the swarm;
3- Update the position of each particle, using its previous position and the updated velocity vector;
4- Go to Step 2 and repeat until convergence.

3.3. Initial distribution

The initial swarm is generally created such that the particles are randomly distributed throughout the design space which is limited by the lower and upper bounds of the design variables. Each considered particle is associated with an analyzed design. In a similar scheme the initial values for particle's velocity are set. In this process, minimum and maximum velocity values for local changes for each particle are predefined parameters of the algorithm.

In the present work, the following equations are used to obtain the random initial position and the velocity vectors.

$$\mathbf{x}_0^i = \mathbf{x}_{\min} + r_1(\mathbf{x}_{\max} - \mathbf{x}_{\min}) \tag{2}$$

$$\mathbf{v}_0^i = \mathbf{v}_{\min} + r_2(\mathbf{v}_{\max} - \mathbf{v}_{\min}) \tag{3}$$

where $\mathbf{x}_0^i$ is the initial position and $\mathbf{v}_0^i$ is the initial velocity for a particle i, in Eq. (2) and Eq. (3), r_1 and r_2 are random numbers between 0 and 1 (they are different for each coordinate), $\mathbf{x}_{\min}$ is the vector of lower bounds for the design variables, $\mathbf{x}_{\max}$ is the vector of upper bounds for the design variables, $\mathbf{v}_{\min}$ is the vector of lower bound of local changes and $\mathbf{v}_{\max}$ is the vector of upper bound of local changes. The velocities values for $\mathbf{v}_{\min}$ and $\mathbf{v}_{\max}$ commonly used in literature are -4 and 4.

3.4. Updating variables

The scheme for updating the position of each particle is shown in Eq. (4).

$$\mathbf{x}_{k+1}^i = \mathbf{x}_k^i + \mathbf{v}_{k+1}^i \Delta t \tag{4}$$

where $\mathbf{x}_{k+1}^i$ represents the position vector of particle i at iteration $k+1$ and $\mathbf{v}_{k+1}^i$ represents the corresponding velocity vector and Δt represent the time step (a unit time step is used throughout the present work), $\mathbf{x}_k^i$ represents the position vector of particle i at iteration k.

The scheme for updating the velocity vector of each particle depends on the particular PSO algorithm under consideration. A commonly used scheme was introduced by Shi and Eberhart (Sobieszczanski-Sobieski and Venter, 2002), as shown in Eq. (5).

$$\mathbf{v}_{k+1}^{i} = w\mathbf{v}_{k}^{i} + c_1 r_1(\mathbf{p}^{i} - \mathbf{x}_{k}^{i}) + c_2 r_2(\mathbf{p}_{k}^{i} - \mathbf{x}_{k}^{i}) \tag{5}$$

where r_1 and r_2 are random numbers between 0 and 1 (they are different for each coordinate), $\mathbf{v}_{k}^{i}$ represent the velocity vector of particle i at iteration k, $\mathbf{p}^{i}$ is the best position found by particle i and $\mathbf{p}_{k}^{i}$ is the best position in the swarm at time k (Sobieszczanski-Sobieski and Venter, 2002).

In optimization context $\mathbf{p}$ is the vector of design variables. As observed in Eq. (5) for each problem there are three parameters to be set, they are the inertia (w) and the two trust parameters (c_1 and c_2). The inertia controls the exploration capability of the algorithm: high values for global behavior while low values are more appropriate for local behavior. The trust parameters indicate how much confidence the current particle has in itself (c_1) and how much confidence it has on the swarm (c_2). For structural optimization problems, in this work, is proposed $w \leq 1,0$ starting the procedure with a value that will be automatically decreased during the optimization process. For the trust parameters, the following values are advised: $c_1 = c_2 = 2$. They are kept constant during the whole procedure.

3.5. Problem parameters update

As mentioned previously, the inertia weight parameter w is adjusted automatically during the optimization. In the present work the changes are based on the coefficient of variation of the objective function values. The goal is to change the w value automatically, with no interaction from the user. A starting value of $w = 0,9$ is used to initially accommodate a more global search and is dynamically reduced to no less than 0. The idea is to terminate the PSO algorithm with a more local search. The w value is adjusted using the updating formula

$$w_{new} = w_{old}\, f_w \tag{6}$$

where w_{new} is the newly adjust w value, w_{old} is the previous w value and f_w is a constant between 0 and 1. In the present work $f_w = 0,975$ is used throughout, resulting in a PSO algorithm with a fairly global search characteristic.

The w value is not adjusted at each design iteration. Instead, the coefficient of variation of objective function values for a subset of best particle is monitored. If the coefficient of variance falls below 1 it is assumed that the algorithm is converging towards an optimum solution and Eq. (6) is applied. A general equation to calculate the coefficient of variance for a set of points is provided in Eq. (7)

$$COV = {StdDev}/{Mean} \tag{7}$$

where COV is the coefficient of variation, $StdDev$ is the standard deviation and $Mean$ is the mean value for the set of points. In this work, a subset of the best 20% of particle from the swarm is monitored.

3.6. Constrained optimization

The basic PSO algorithm is defined for unconstrained problems only. Since most engineering problems are constrained in one way or the other, it is important to add the capability of dealing with constrained optimization problems. It was decided to deal with constraints by making use of a quadratic exterior penalty function. This technique is often used to deal with constrained problems in genetic algorithms (Goldberg, 1989). In the current implementation, when one or more of the constraints are violated, the objective function is penalized as shown bellow.

$$\mathrm{P}(\mathbf{x}) = \bar{f}(\mathbf{x}) + \mu\alpha(\mathbf{x}) \tag{8}$$

where μ is the parameter of penalty, $\bar{f}(\mathbf{x})$ is the normalized objective function of problem given as:

$$\bar{f}(\mathbf{x}) = f(\mathbf{x}) / f(\mathbf{x}_0) \tag{9}$$

and $\alpha(\mathbf{x})$ is given as:

$$\alpha(\mathbf{x}) = \sum_{i=1}^{n_{con}} \max\left[0; g_i(\mathbf{x})\right]^2 \tag{10}$$

and $g(\mathbf{x})$ is the set of all constraints (with violated constraints having values larger than zero). In the present work a penalty parameter of $\mu = 10^8$ is used during the whole procedure.

3.7. Treatment for particles with violated constraints

A simple modification to Eq. (5) is proposed when there are particles with one or more violated constraints. The modification can be explained by considering particle i, which assumed to have one or more violated constraints at iteration k. By re-setting the velocity vector of particle i at iteration k to zero, the velocity vector at iteration $k+1$ is obtained as

$$\mathbf{v}_{k+1}^i = c_1 r_1 (\mathbf{p}^i - \mathbf{x}_k^i) + c_2 r_2 (\mathbf{p}_k^i - \mathbf{x}_k^i) \tag{11}$$

3.8. Convergence criterion

A convergence criterion is necessary to avoid any additional function evaluations after an optimum solution is found. In the present work a particle cluster formation as a stop criterion was created and implemented. The cluster formation consists in concentration of the particles. The procedure is locked up if a considerable amount of particles are grouped around of the same point.

For a cluster formation check it is necessary to select an amount of particles with the best positions in the design space. Then it is verified the relative position between them. When the greater difference between all distances is smaller than a tolerance, previously determined, then the convergence is achieved. The number of pre-selected particles considered in the cluster formation has been studied.

4. PRESENT APPLICATION

The application addressed in this work consists in plane truss submitted to static loads. The global equation of the specific problem is given as

$$\mathbf{K}\boldsymbol{d} = \boldsymbol{F} \tag{12}$$

where $\mathbf{K}$ is the global stiffness matrix, $\boldsymbol{d}$ is the displacement vector (unknowns) and $\boldsymbol{F}$ is the force vector applied.

One of the outputs with great interest in structural designs are the stress on the truss bars, calculated for each bar as follows

$$\sigma^e = f^e / A^e \tag{13}$$

in which A^e is the area of one generic bar e and f^e is obtained as follows.

From a local equilibrium we have

$$F^e = K^e d^e \tag{14}$$

and

$$fe^e = \mathbf{R}^T F^e \tag{15}$$

in which case $\mathbf{R}$ is the matrix of rotations, see (Cook, 1981), and f^e of Eq. (13) could be, for instance the first component of vector fe^e, i. e.,

$$f^e = fe^e(1) \tag{16}$$

5. RB APROACH

5.1. Central idea

To construct an approximation for the displacements and consequently to any solution output (here we will focus on the compliance) satisfying efficiency and accuracy requirements.

5.2. Efficiency

The use of an affine decomposition to the stiffness matrix of the conventional (costly) problem is the requirement to perform inexpensive computational calculations. Thus we rewrite $\mathbf{K}$ as

$$\mathbf{K}(\boldsymbol{\mu}) = \sum_{r=1}^{R} \mu_r \mathbf{K}_r \tag{17}$$

in which each parameter μ_r is a cross sectional area to be specified for different geometric regions r ($r = 1,\ 2,\ ...,\ R$) of the truss. It is important to emphasize that the stiffness matrix $\mathbf{K}_r$ is restricted to particular geometric region r and is independent of $\boldsymbol{\mu}$.

5.3. Approximation

To construct the RBM approximations, firstly a sample S^N in the design D must be chosen

$$\mathrm{S}^N = \left\{ (\mu_1,\ ..., \mu_R)^1,\ ..., (\mu_1,\ ..., \mu_R)^N \right\} \tag{18}$$

where each $(\mu_1,\ ..., \mu_R)^i$ is in D, this means

$$\mu^l \le \mu_r \le \mu^{up} \tag{19}$$

in which μ^l and μ^{up} are the lower and upper limits respectively of D. The associated reduced basis space is represented by

$$W^N = span\left\{ \boldsymbol{\zeta}^i,\ i = 1,\ ...,\ N \right\} \tag{20}$$

in which

$$\boldsymbol{\zeta}^i = \boldsymbol{d}\left((\mu_1,\ ..., \mu_R)^i \right),\ i = 1,\ ..., N \tag{21}$$

Taking the above definitions, the reduced-basis approximation problem can be formulated as:

For $\boldsymbol{\mu} \in D$ find $\boldsymbol{d}^N(\boldsymbol{\mu})$ in the Galerkin projection of $\boldsymbol{d}(\boldsymbol{\mu})$ onto W^N. Therefore, taking into account the observations made previously, the reduced-basis approximation $\boldsymbol{d}^N(\boldsymbol{\mu})$ is expressed as

$$\boldsymbol{d}^N(\boldsymbol{\mu}) = \sum_{i=1}^{N} \alpha^i(\boldsymbol{\mu}) \boldsymbol{\zeta}^i,\ \alpha^i(\boldsymbol{\mu}) \in \Re^N \tag{22}$$

The above equation means that $\boldsymbol{d}^N$ can be expressed as a linear combination of the displacements solutions $\boldsymbol{\zeta}^i$. In matrix form Eq. (22) is rewritten as

$$\boldsymbol{d}^N(\boldsymbol{\mu}) = \mathbf{Z}\boldsymbol{\alpha}(\boldsymbol{\mu}) \tag{23}$$

Using stationary conditions to the total potential energy and Eq. (23) to represent the displacements field, we end up in the following equation (Prud'homme et al, 2002)

$$\mathbf{K}^N(\boldsymbol{\mu})\boldsymbol{\alpha}(\boldsymbol{\mu}) = \boldsymbol{F}^N \tag{24}$$

in which

$$\mathbf{K}^N(\boldsymbol{\mu}) = \mathbf{Z}^{\mathrm{T}}\mathbf{K}(\boldsymbol{\mu})\mathbf{Z} \in \Re^{N\times N} \tag{25}$$

and

$$\boldsymbol{F}^N = \mathbf{Z}^{\mathrm{T}}\boldsymbol{F} \in \Re^N \tag{26}$$

5.4. Accuracy

The accuracy checking for the method is accomplished due to consideration of a posteriori error estimator procedure. To compute the error in a computationally efficient way the following residual equation should be solved

$$\mathbf{C}(\boldsymbol{\mu})\hat{\boldsymbol{e}}(\boldsymbol{\mu}) = \mathrm{R}(\boldsymbol{\mu}) \tag{27}$$

in which $\mathbf{C}(\boldsymbol{\mu})$ is a symmetric operator defined such that

$$\boldsymbol{e}^{\mathrm{T}}(\boldsymbol{\mu})\mathbf{K}(\boldsymbol{\mu})\boldsymbol{e}(\boldsymbol{\mu}) \le \hat{\boldsymbol{e}}^{\mathrm{T}}(\boldsymbol{\mu})\mathbf{C}(\boldsymbol{\mu})\hat{\boldsymbol{e}}(\boldsymbol{\mu}) \tag{28}$$

In the above equations $\boldsymbol{e}(\boldsymbol{\mu})$ is the exact error and $\hat{\boldsymbol{e}}(\boldsymbol{\mu})$ is the approximated error and R is the residue. To compute $\mathbf{C}$ we consider the so-called point conditioner strategy (Prud´homme et al, 2002):

$$\mathbf{C}(\boldsymbol{\mu}) = g(\boldsymbol{\mu})\hat{\mathbf{K}} \tag{29}$$

in which $g(\boldsymbol{\mu}) = \min\{\mu_r\},\ r = 1,\ ..., R$ and

$$\hat{\mathbf{K}} = \sum_{r=1}^{R}\mathbf{K}_r \tag{30}$$

It is important here to emphasize that the solution and error computations are very fast as it take the advantage of using several precomputed quantities ($\boldsymbol{\mu}$ independent). This is possible due the Eqs. (24, 25) together with the decomposed form of our stiffness matrix, defined in Eq. (17).

6. EXAMPLES

The approaches presented before are tested for some truss applications. In this work some benchmark examples are analyzed. The examples highlight the advantage of using the RBM over the conventional approach as problem complexities increases. For comparison purposes, a fully FE based solution (Afonso and Patera, 2003, Albuquerque, 2005) are also considered.

6.1. Geometry definition

Three different benchmark trusses are considered. They are: ten bars truss, two hundred bars truss and nine hundred and thirty bars truss. Their geometry and loading definition are indicated in Fig. (1). The Elastic Modulus is $E = 2,07\times10^{11}$ for all cases.

For the reduced-basis solution, each truss is subdivided in three regions, regions I, II and III, indicated in Fig. (1), in which the struts are determined to have different cross sectional areas. This leads to R=3 in the present study. The initial area $(A_1,\ A_2,\ A_3)$ of each one of the two trusses is the same. It is considered $N = 9$ and the sample S_1^N for the ten bars truss and $N = 15$ and the sample S_2^N for the remaining two trusses. The samples are indicated in Tab. (1) and Tab. (2) respectively. The quantity of interest to be computed is the stress in all their members.

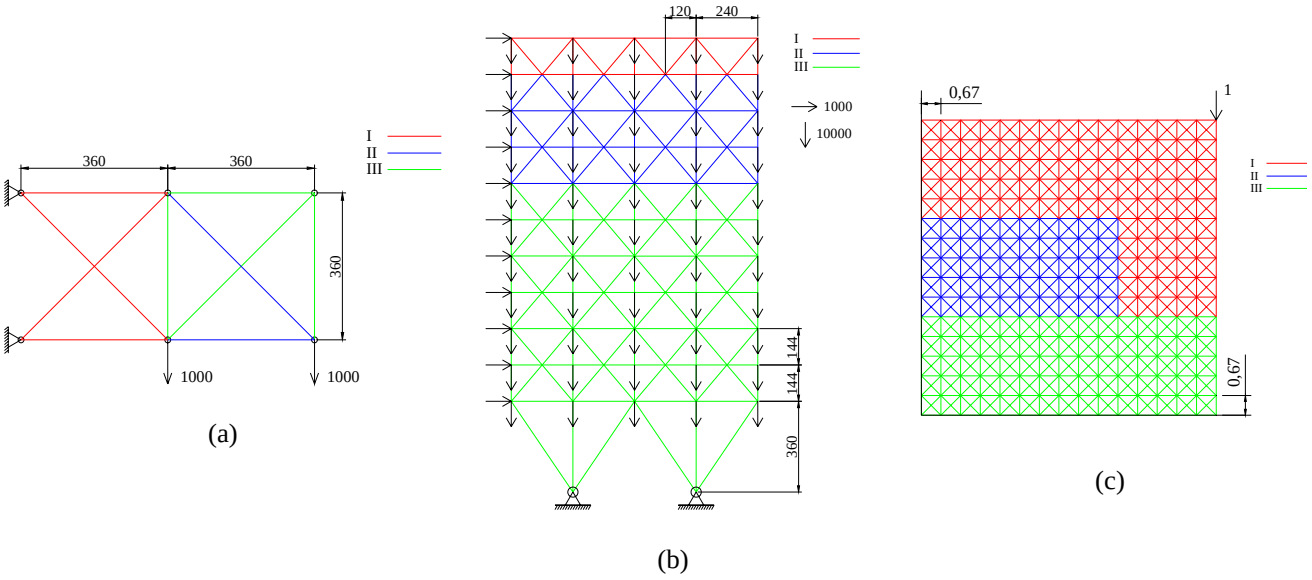

Figure 1. Geometry and loading definition for: (a) ten bars truss, (b) two hundred bars truss and (c) nine hundred and thirty bars truss

Table 1. Sampling S_1^N for the ten bars truss example sample.

N	1	2	3	4	5	6	7	8	9
μ_1	0.1	1.0	10.0	1.0	1.0	10.0	0.1	1.0	10.0
μ_2	0.1	0.1	0.1	2.0	5.0	2.0	10.0	10.0	5.0
μ_3	0.1	0.1	0.1	5.0	1.0	1.0	10.0	10.0	10.0

Table 2. Sampling S_2^N for the two hundred bars truss and nine hundred and thirty bars truss examples sample.

N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
μ_1	6,62	2,89	9,77	3,94	7,47	8,16	2,53	5,50	6,90	0,85	1,50	8,70	5,20	4,63	0,46
μ_2	9,62	1,93	4,79	2,12	7,64	8,28	9,25	6,27	0,90	5,99	7,10	2,81	4,51	0,21	3,60
μ_3	7,43	2,36	6,12	8,65	0,70	4,11	3,77	9,88	9,28	7,16	1,02	1,79	2,92	4,73	5,46

6.2. Optimization

The PSO algorithm presented in this work is used in this section to obtain optimum truss designs. For PSO solution the population in the swarm is 50 particles. The other parameters are those previously mentioned.

The volume is the objective function to be minimized. A part from that, the design variables are $\mu_1 = A_1$, $\mu_2 = A_2$ and $\mu_3 = A_3$ and the design space considered is $D = [0.1,\ 10.0] \times [0.1,\ 10.0] \times [0.1,\ 10.0]$. This means that their lower and upper limits are respectively 0.1 and 10.0. Apart from those, stress constraints are imposed to the problem, such that the allowable stress value is given in Tab. 3 to all bars for each truss.

Table 3. Stress constraints.

	Tension	Compression
10 bars truss	4000	4000
200 bars truss	20000	20000
930 bars truss	0,5	0,5

Tables 4, 5 and 6 show the optimization results for the 10 bars truss example, the 200 bars truss and 930 bars truss respectively, considering both strategies investigated here (FE and RBM).

Table 4. 10 bars truss example results

	FE solution	RBM solution
μ_1	0,52	0,52
μ_2	0,29	0,29
μ_3	0,10	0,10
Volume	$1,318\times10^3$	$1,318\times10^3$
Time (normalized)	0,0314	1,00

Table 5. 200 bars truss example results

	FE solution	RBM solution
μ_1	0,43	0,43
μ_2	1,51	1,51
μ_3	7,27	7,27
Volume	$1,726\times10^5$	$1,726\times10^5$
Time (normalized)	2,719	1,00

Table 6. 930 bars truss example results

	FE solution	RBM solution
μ_1	1,56	1,56
μ_2	0,1	0,1
μ_3	0,24	0,24
Volume	$5,88\times10^2$	$5,88\times10^2$
Time (normalized)	4,269	1,00

For each case, both strategies converge to the same optimum. However, the fast computation inherit to RBM is perceived for the 200 bars and 930 bars cases. Also the difference in CPU time using both schemes increases as the structure gets more complex. For the RBM the CPU time is almost constant as the number of degrees of freedom increases. For the ten-bar truss N is greater than the total number of degrees of freedom. Therefore, as the structure of $\mathbf{K}^N$ matrix is dense this explains the CPU time consumed for that case be higher for RB solution when compared with FE solution. This highlights the importance of the use RBM for the design of large structures.

7. CONCLUSIONS

Optimum designs were here obtained for classical trusses problems. The RBM was integrated in a PSO algorithm in order to conduct fast computations. A certify of fidelity for the reduced basis was obtained through the implementation of a posteriori error estimator.

The results were compared to the conventional PSO, which employs FE method. As the complexity of the FE's equation increases the advantage of using the reduced basis in the SSO algorithm was highlighted.

8. ACKNOWLEDGEMENTS

The authors acknowledge the financial support given by CNPq, FACEPE and UFPE to the execution of the present work.

9. REFERENCES

Afonso, S.M.B. 1995, "Shape optimization of Mindlin-Reissner shells under static and free vibration conditions", PhD thesis. University of Wales Swansea, Swansea, Wales, UK.

Afonso, S.M.B., Azevedo, A.A., 2000, "Optimum Solutions for Trusses Using Sequential Quadratic Programming and Genetic Algorithms", CONEM, I Congresso Nacional de Engenharia Mecânica, Anais (CD-ROM). Natal, RN: UFRN.

Afonso, S.M.B., Patera, A.T. 2003 "Structural optimization in the framework of reduced basis method", CILAMCE 2003 XXIV Iberian Latin American Congress on Computational Methods in Engineering (held in Ouro Preto-MG, Brazil, 2003), CD-ROM, pp 1-15.

Albuquerque, T.M.M., Afonso, S.M.B., Lyra, P.R.M. 2005, "Fast optimization procedure for heat transfer applications", CILAMCE 2005 XXVI Iberian Latin American Congress on Computational Methods in Engineering (held in Guarapari-ES, Brazil, 2005) CD-ROM, pp 1-14.

Almroth, B.O., Stern, P., Brogan, F.A. 1978, "Automatic choice of global shape functions in structural analysis", AIAA J. 16: 525-528.

Bochenek, B., Forys, P., 2003, "A New Particle Swarm Optimizer and Its Application to Structural Optimization", *Institute of Applied Mechanics, Cracow University of Technology.*

Cook, R. D., 1981, *"Concepts and Applications of Finite Elements Analysis", 2nd ed., John Wiley & Sons, Singapore.*

Fink, J.P., Rheinboldt, W.C. 1983, "On the error behavior of the reduced basis technique for nonlinear finite element approximations", Z. Angew. Math. Mech. 63: 21-28.

Goldberg, D. E., 1989, "Genetic Algorithms in Search, Optimization, and Machine Learning".

Haftka, R., van Keulen, F. 2004, "Special issue on approximation in optimization of structural and multidisciplinary optimization", Structural and Multidisciplinary Optimization Journal 27.

Noor, A.K., Peters, J.M. 1980, "Reduced basis technique for nonlinear analysis of structures". AIAA J. 18: 455-462.

Pomeroy, P. 2003, "An Introduction to Particle Swarm Optimization - <http://www.adaptiveview.com/articles/ipsoprnt.html>

Prud'homme, C., Rovas, D.V., Veroy, K., Machiels, L., Maday, Y., Patera, A.T., Turicini, G. 2002, "Reliable real-time solution of parameterized partial differential equations: reduced-basis output bound method", Journal of Fluids Eng: 124: 70-79.

Venter, G. and Sobieszczanski–Sobieski, J., 2002, "Particles Swarm Optimization", AIAA– 1235, p. 1–9.

Vanderplaats, G. N., 1999, "Numerical Optimization Techniques for Engineering Design", Vandersplaats Research and Development, Inc., 3rd ed.

10. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.