MODERN BUILDING
MATERIALS, STRUCTURES
AND TECHNIQUES

May 19–21, 2010, Vilnius, Lithuania
The 10th International Conference

Faculty of Civil Engineering
Vilnius Gediminas Technical University
Saulėtekio ave. 11, LT-10223 Vilnius, Lithuania
Phone +370 5 2745243, Fax +370 5 2745016
E-mail: *statkon@vgtu.lt*

http://www.vgtu.lt/en/editions/proceedings

GLOBAL OPTIMIZATION FOR SOME ENGINEERING PROBLEMS: DIFFICULTIES AND PROSPECTS

Rimantas Belevičius[1], Dmitrij Šešok [2]

[1]*Vilnius Gediminas Technical University, Saulėtekio ave. 11, LT-10223 Vilnius, Lithuania.*
E-mail: rimantas.belevicius@vgtu.lt
[2]*Institute of Mathematics and Informatics, Akademijos str. 4, LT-08663 Vilnius, Lithuania.*
E-mail:dsesok@gmail.com

Abstract. On the basis of one engineering non-convex optimization problem – optimization of pile placement schemes in grillage type foundations, several global optimization algorithms are compared. Comparison reveals clear advantages of stochastic algorithms, among which the simulated annealing is most promising. The prospects of using simulated annealing on different computer platforms are shown; the largest solved problem contained 55 decision variables.

Keywords: global optimization, pile placement schemes, simulated annealing, clustering, GRID computations.

Introduction

Optimization is an inherent part of all engineering practice. In the construction of buildings means that, all parts of buildings from foundations to roofs should be designed and built optimally and thrifty as much as the conditions of safety and comfort allow. Mathematically, a number of such optimization problems (e.g., topology optimization of roof-supporting truss systems, load carrying structures of bridges, optimization of pile placement schemes under grillage-type foundations or under pile caps, etc.) can be reduced to global minimization of multimodal functions. Such problems are difficult in the sense of algorithmic complexity, and global optimization algorithms are computationally very expensive. Aim of this paper is to review the capabilities and application possibilities of different global optimization algorithms as well as techniques in engineering practice. Here the optimization of pile placement schemes serves as a really good benchmark since the problem is non-convex, highly sensitive to the design parameters (i.e., positions of piles under the connecting beams), and the global solution (or at least the lower bound of solution) is always known in advance. Several deterministic and stochastic global optimization algorithms without/and including some heuristic information on the problem are compared. The prospects of an industrial-scale optimization of foundation schemes on a single PC, clusters of computers, and distributed computing on a GRID are shown.

The grillage-type foundations are ones of the most popular and effective schemes of foundations, especially in case of weak grounds. The optimal scheme of grillage should possess the minimum possible number of piles. Theoretically, reactive forces in all piles should approach the limit magnitudes of reactions for those piles. This goal can be achieved by choosing appropriate pile positions. However, some problem constraints such a minimum allowable distance between adjacent piles (due to the capabilities of pile driver), or so-called immovable piles which are introduced into certain positions of pile scheme by the analyst and do not participate in the optimization process, may hinder to achieve the global solution. In all numerical examples of the present paper the immovable piles are not dealt with.

Whereas the beam optimization problems in form of optimal sizing of beams in grillage structures under given boundary and loading conditions (see, e.g. (Chamoret *et al.* 2008) and references therein) or optimal layout of grillages (e.g. (Rozvany 1997)) attracted lot of attention, only a few papers deal with the optimization of pile placements schemes. In (Kim *et al.* 2005) an optimal pile placement scheme under the raft is sought that minimizes the differential settlements of the raft using genetic algorithms (GA). In (Chan *et al*. 2009) the pile groups under the solid pile cap together with the cap dimensions are optimized using hybrid GA. The local search algorithms were employed for optimal placing of piles under a separate beam of grillage (Belevičius and Valentinavičius 2001) and under whole grillage using iterative algorithm

on the basis of mentioned work (Belevičius *et al.* 2002). Experience shows that the objective function for practical grillage optimization problems possesses many local minima points. Due to this the local search obviously is not a proper choice, and global optimization algorithms are the necessity. The deterministic global optimization algorithms proved to require non-realistic computer resources for even small-scale grillage optimization problems (Čiegis *et al.* 2006). Promising results for larger-scale grillages (up to tens of design variables) were achieved with stochastic algorithms: GA (Belevičius and Šešok 2008), simulated annealing (SA) algorithm (Šešok *et al*. 2010a; Šešok *et al*. 2010b).

The paper is organized as follows. In the next chapter the mathematical formulation of the optimization problem is presented. Chapter 3 provides comparison of different optimization algorithms solving 10 real-life grillages. In the 4th chapter attention is given to the computing on different computer platforms. Finally, some general conclusions on the prospects of application of global optimization algorithms to the optimization of grillages are drawn.

Problem formulation

The optimization problem is formulated as follows:

$$f^* = \min_{x \in D} f(x) \tag{1}$$

where $f(x)$ is a nonlinear objective function of continuous variables $f : \Re^n \to \Re$, n is the number of design parameters x defining positions of piles, $D \subset \Re^n$ is a feasible region of design parameters. Besides of the global minimum f^* one or all global minimizers $x^* : f(x^*) = f^*$ should be found. No assumptions on unimodality are included into formulation of the problem – many local minima may exist.

In this paper the maximal vertical reactive force at a pile is considered as the objective function:

$$f(x) = \max_{i=1,\ldots,N_a} R_i(x) \tag{2}$$

where N_a is the number of piles, $R_i(x)$ is the reactive force at i-th pile.

Since a supporting pile may reside only under connecting beams, there are evident restrictions on the positions of piles: during the optimization process the piles can move only along the connecting beams. Therefore, a two-dimensional beam structure of the grillage is "unfolded" to a one-dimensional construct, and the piles are allowed to range through this space freely.

In such a formulation one design parameter corresponds to a position of one pile in the one-dimensional construct ($n=N_a$). The backward transformation restores the positions of piles in the two-dimensional beam structure of the grillage. The constraints for the design parameters are as follows:

$$0 \le x_i \le L, \quad i = 1,\ldots, N_a \tag{3}$$

where x_i is a design parameter defining the position of i-th pile. L is the total length of all beams in the grillage. If the minimal possible distance δ between adjacent piles is specified, there are additional constraints

$$\|x_i - x_j\| \ge \delta, \; i \ne j \tag{4}$$

where x_i are two-dimensional coordinates of piles and $\|x_i - x_j\|$ denotes the distance between piles. To cope with this constraint a penalty is included in the objective function.

A finite element program is used as a "black-box" routine to the optimization program for solution of direct problem to find reactive forces in the grillage. In the direct problem the connecting beams in the grillage are idealized as the beam elements, while the piles are treated as supports, i.e. finite element mesh nodes with given elastic boundary conditions. Since time of optimization crucially depends on the solution time of the direct problem, fast problem-oriented original FORTRAN programs with a special mesh pre-processor have been developed. The finite element and optimization programs interact in the following way: optimization program produces a guess for pile positions in one-dimensional construct; the finite element program transforms it to the two-dimensional grillage, automatically subdivides the grillage into finite element mesh, solves the statical problem and obtains the objective function – maximum reactive force among all piles. On basis of this value the optimization algorithm produces a new guess, and the loop is repeated.

The beam elements have 2 nodes with 6 degrees of freedom each (3 displacements along the coordinate axes and 3 rotations about these axes). The stiffness matrix for element can be found in many textbooks, e.g. by (Zienkiewicz and Taylor 2005).

The main statics equation is

$$[K]^a \{u\}^a = \{F\}^a \tag{5}$$

where a stands for the ensemble of elements (not shown in eq. below), $\{u\}$ are the nodal displacements, and $\{F\}$ are the active forces. The reactive forces at piles are available after obtaining the nodal displacements:

$$R_i = \sum_j [K_{ij}] u_j \tag{6}$$

The finite element program requires the following initial data:

- The geometrical scheme of connecting beams;
- Cross-section data of all beams (area, moments of inertia);
- Material data of all beams (material in one beam is treated as isotropic);
- Positions of immovable piles (if any);
- Maximum allowable reactive force at any pile R_{allw};
- Minimum possible distance between adjacent piles;
- Stiffnesses of pile (vertical, rotational);

– Loading data. Active forces can be applied in the form of concentrated loads and moments at any point on beam, or in the form of distributed trapezoidal loadings at any segment of beam.

Comparison of optimization algorithms

The algorithms were compared using the results of optimization of pile placement schemes of 10 practical grillages. All these grillages are of small-to-medium scale, requiring from 17 to 55 piles. Data for these problems are obtained from several Dutch design bureaus (courtesy of Consultancy W. F. O. B.V., Paauw B. V. Aannemingsbedrijf, and others) which use the analysis and design package *MatrixFrame*. *MatrixFrame* has some capabilities for optimization of pile placement schemes employing local search methods (Belevičius and Valentinavičius 2001). One trait is common to all these problems: the current optimization routine of *MatrixFrame* was not capable to yield even a rational scheme of pile placement.

The typical grillage is shown in Fig 1. All the characteristics of all 10 grillages are summarized in Table 1. There are no immovable piles, therefore the numbers of piles N_a and design variables n coincide. R_{allw} denotes the allowable reactive force at the piles, it is equal for all piles in all problems. The number of piles is obtained dividing the total sum of loadings by the allowable reactive force of piles – the number of piles cannot be less than this number. R_{ideal} denotes the ideal theoretical value; the ideal solution is when all reactive forces at the piles are equal. In all these problems the proportion between the total loading and the allowable reaction is such that the engineering solution requires achieving almost the ideal solution. This is the main reason, why these problems are difficult to solve.

Generally, all global optimization methods can be categorized into non-gradients methods which require information from the objective function only, and gradient methods which beside the information about objective function use in addition the gradient or higher derivative information. Accurate calculation of function gradients is usually computationally expensive and problematic. Also, the algorithms may be divided into deterministic algorithms which guarantee the same solution on every run of algorithm, and stochastic ones. The last algorithms produce different results on every run therefore the decisions about achieved solutions can be made only on the basis of several tens of runs.

Algorithms involved in comparison cover all this nomenclature. To make a fair comparison, the total number of objective function evaluations for each algorithm is the same – N = 5000. The following algorithms were used:

– Modified random search (MRS)
– Simulated annealing (SA)
– Genetic algorithm (GA)
– Simplex method (SM)
– Variable metric method (VM)
– NEWUOA algorithm

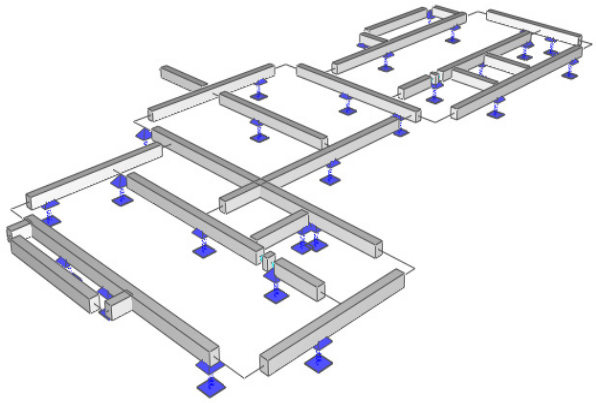

Fig 1. Grillage No 1 (according to the Table 1)

Table 1. Characteristics of grillages

Problem No	N_a	L	R_{allw}	R_{ideal}
1	25	172.9	325	307.47
2	18	52.9	110	104.12
3	31	84.1	105	101.85
4	31	84.9	105	101.24
5	30	63.9	100	97.51
6	37	80.1	100	97.53
7	23	129.1	300	287.35
8	34	137.9	250	236.28
9	17	97.6	250	244.71
10	55	315.61	350	349.05

MRS is a random search algorithm, where the decision variables are generated randomly with uniform distribution, but with one additional constraint: the difference between two decision variables should be larger than some given distance. If the constraint is violated, the structure is considered to be non-feasible. Such a heuristic modification is motivated by the fact that due to the usual distribution of loading over the grillage beams, the piles also should be spread over the whole space of grillage. Reactive forces at the piles are found for N feasible random structures and one with the smallest maximal reactive force at a pile is considered to be the best solution found.

SA is also a stochastic algorithm which was invented for simulation of metal annealing process (Kirkpatrick *et al.* 1983). Here the current solution is replaced by a random solution with a probability that depends on the difference of the function value and the so-called temperature parameter. In the beginning the temperature parameter is large allowing non-improving changes. Gradually temperature is decreased and the search becomes descent. In our implementation the initial solution is obtained using MRS taking the best feasible solution out of *N_init*=200. Then SA algorithm is employed.

Stochastic GAs (Goldberg 1989; Šešok *et al.* 2008) simulate evolution (selection, mutation, crossover) in which a population of solutions evolves improving values of the objective function. In our implementation, the initial population of *popsize* individuals is generated using

MRS. The new generation is obtained from the previous one using selection, crossover and mutation operations. Besides the population size and generation number, results of algorithm depend on the probabilities of crossover and mutation, and the selection strategy. Here all values of these parameters are chosen numerically, on the basis of numerous experiments.

The SM is a general method for optimization of nonlinear multidimensional function requiring only function evaluations, but not derivatives. A simplex is the geometrical figure consisting of $n + 1$ vertices in n dimensions. The simplex method takes a series of steps, moving the vertex of the simplex where the objective function is worst. We use implementation of the method form (Press 1992).

VM is based on the Newton's method to find the stationary point of objective function where its gradient is 0. Here the BFGS form of method (Press 1992) is used. Since variable metric method requires gradient information, the results of sensitivity analysis are used. The starting point is the best solution obtained using MRS in the given number of iterations *N_init*. Then the local search is performed. Initialization and local search is repeated until the overall number of objective function evaluations reaches *N*.

The NEWUOA algorithm is iterative algorithm for nonlinear optimization. A quadratic model is used in a trust region procedure for adjusting the variables (Powell 2006). The quadratic model interpolates the function at $2n+1$ points, only one interpolation point is altered on each iteration. The starting point is the best solution obtained using MRS in the given number of iterations *N_init*. The NEWUOA algorithm is stopped when the number of objective function evaluations reaches *N*.

More thorough description of algorithms is given in (Šešok *et al.* 2010a, 2010b). Parameters for all algorithms are summarized in Table 2. Other parameters, for example the initial temperature for SA or probabilities of crossover and mutation for GA are tuned to the problems under consideration (Šešok *et al.* 2010a, 2010b). All algorithms are initialized using MRS. This heuristic modification significantly improves the results and enables optimal design of real grillages. If a pure random search is used for initialization, the results are much worse.

Table 2. Parameters for algorithms

Algorithm	Parameters
MRS	$N = 5000$
SA	$N\_init = 200$, $N = 5000$, $t_1 = 5$, $t_2 = 2$
GA	$popsize = 20$, $p\_cross = 0.9$, $p\_mut = 0.3$, $G = N/popsize = 5000/20 = 250$
SM	$N\_init = 300$, $N = 5000$
VM	$N\_init = 300$, $N = 5000$
NEWUOA	$N\_init = 300$, $N = 5000$

The algorithms were tested on a personal computer with Intel(R) Xeon(R) CPU E5420 @ 2.50GHz, 3069 MB RAM, 32-bit Operating System. 28 independent runs were performed for each algorithm.

Table 3 shows the average results of all 28 runs. In Table 4 the best obtained results are rendered. The average times of one run are given in Table 5. In all tables the bold type highlights the best achieved results. The last column in Tables 3 and 4 presents discrepancies between the best obtained results and the corresponding ideal value.

Thus, an ideal solution was not obtained for any problem in any of 28 independent runs; in each run the objective function was evaluated 5000 times. The best objective values found differ from ideal ones from 2.2 % (Problem 2, 18 piles) to 46.8 % (Problem 8, 34 piles), while the average results – from 4.8 % to 63.4 % for the same problems. As it can be expected, generally discrepancy is larger for problems with the larger number of piles.

Table 3. Average of the best values found in 28 runs

Problem No	R_{ideal}	RS	MRS	SA	GA	SM	VM	NEWUOA	Discrepancy, %
1	307.47	593.33	470.90	**371.55**	405.36	486.50	454.01	394.92	20.8
2	104.12	153.07	125.71	**109.10**	112.49	131.75	127.64	113.35	4.8
3	101.85	258.45	144.46	**119.06**	124.48	153.82	143.60	119.80	16.9
4	101.24	265.41	141.18	117.10	123.72	147.46	139.82	**116.81**	15.4
5	97.51	318.16	126.08	**106.25**	112.23	133.28	128.13	108.68	9.0
6	97.53	460.31	160.18	132.22	144.16	172.38	160.85	**132.07**	35.4
7	287.35	472.74	379.80	**314.11**	332.11	402.36	382.45	330.16	9.3
8	236.28	695.60	494.83	413.57	444.88	520.98	491.25	**385.99**	63.4
9	244.71	402.17	343.91	**281.12**	294.62	369.06	334.90	292.28	14.9
10	349.05	1321.48	702.53	562.79	636.87	759.33	705.57	**559.88**	60.4

Table 4. The best values found in 28 runs

Problem No	R_{ideal}	RS	MRS	SA	GA	SM	VM	NEWUOA	Discrepancy, %
1	307.47	503.93	430.67	**339.30**	360.73	436.68	429.34	370.79	10.4
2	104.12	136.23	119.16	**106.36**	106.52	125.83	123.18	107.18	2.2
3	101.85	181.75	138.37	**107.25**	115.04	138.58	135.54	109.00	5.3
4	101.24	175.33	131.15	**106.80**	112.10	132.10	133.65	108.16	5.5
5	97.51	170.61	116.88	102.00	104.58	120.42	120.74	**101.05**	3.6
6	97.53	364.21	148.24	**117.26**	119.65	151.95	146.01	125.37	20.2
7	287.35	409.73	351.71	**298.11**	310.24	355.82	358.73	306.31	3.7
8	236.28	538.04	440.42	357.67	363.15	426.74	430.11	**346.94**	46.8
9	244.71	354.98	318.38	**253.00**	274.55	339.68	313.40	270.68	3.4
10	349.05	1026.48	670.72	**463.34**	519.07	698.15	648.84	486.46	32.7

Table 5. Average solution times of one experiment, sec

Problem No	N_a	RS	MRS	SA	GA	SM	VM	NEWUOA
1	25	899	943	960	844	**821**	855	878
2	18	92	96	91	**82**	90	91	91
3	31	835	876	822	**732**	811	801	753
4	31	1015	1045	1046	935	**910**	914	916
5	30	319	319	302	**276**	313	318	315
6	37	811	823	742	718	**674**	761	750
7	23	649	695	683	635	**622**	660	656
8	34	3009	2982	2956	2668	2467	**2455**	2489
9	17	424	417	410	**359**	413	419	423
10	55	2198	2136	2125	1934	1826	**1811**	1850

The comparison of algorithms clearly shows the advantages of SA with tuned parameters both for best obtained as well as for average results of 28 runs. Comparing the average results, only the NEWUOA significantly outruns the SA for Problem 8 (34 piles). It shows slightly better results also for Problems 4 (31) and 6 (37). As to the best results, the NEWUOA obtains the best solutions for two problems. Nearly in all cases the GA shows the third best solutions; it never outruns the SA or NEWUOA algorithms. The results of SM and VM are sometimes even worse than that of MRS what means that it is not worth to spend time to find local minima for these problems, but rather to search wider globally since local searches do not improve the value of the objective function significantly.

One numerical experiment takes from approximately 1.5 (algorithm GA, Problem 2, 18 piles) to 41 minutes (VM, Problem 8, 34 piles). Comparing the timings of all algorithms, the clear winners here are the GA and SM. However, the other algorithms are not far behind, until ~18 % in ultimate cases. Thus, the timings can not be treated as the decisive factor for comparison of these algorithms.

All these results were obtained evaluating the objective function for 5000 times and repeating the experiment 28 times. It can be expected, that longer computations may render the relevant solutions. On the basis of this, the SA algorithm and the largest optimization problem – grillage No 10 with 55 piles, for which the discrepancy between average and ideal solutions reaches ~60 %, were chosen for further experiments with massive computation systems.

Comparison of solutions on different computer platforms

According to (Mockus 1967), the convergence rate of stochastic algorithm is asymptotic and slow, not more than $log(N)$, where N is the number of iterations. Theoretically, in order to decrease the discrepancy between obtained optimization result and the global solution twice, we have to perform about 100 times more iterations. Thus, for the 55-pile grillage the number of iterations of $\sim 10^7$ should guarantee the discrepancy between obtained and ideal solutions of around 10 %. Since one run of SA with 5000 iterations for 55-pile grillage takes 2125 seconds on PC (Belevičius *et al.* 2010) and at least several tens of independent runs are needed, such a number of iterations is inconceivable for PC platform.

SA and GA algorithms ideally suit for parallel computations, because one run of an algorithm can be directly allotted for one processor. Any changes are not needed for an algorithm, if the intrinsic parallelization of the task (e.g. in the solution of a system of linear equations) is not

involved; data exchange between processors is not needed in this case. Two computer platforms were used for the solution: GRID and PC cluster.

GRID computing. Solution scheme in GRID is shown in Fig 2. Each processor obtains its task, which is completely independent from the others. Each process reads its data files and produces its result files in ASCII format. The produced files are not large therefore it is not necessary to store them in storage elements. At the end of computations result files are automatically processed in order to sort numerical solutions, considering the computed values of objective function, and to store the best results. It is quite difficult to handle large amount of tasks and result files, therefore good software including graphical user interface is required for automatic job submission, monitoring and result processing.

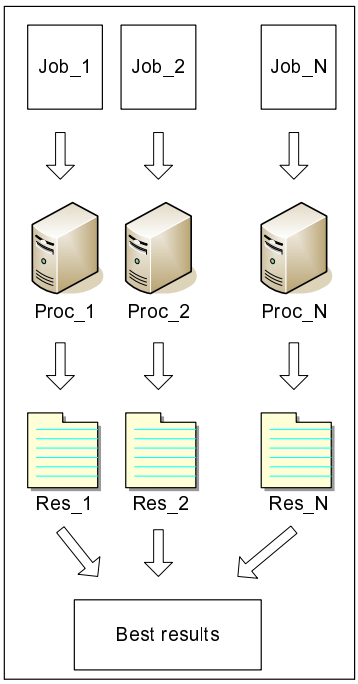

Fig 2. Solution scheme in GRID

Grillage optimization software has been deployed in BalticGrid infrastructure by using Gridcom (Šešok *et al.* 2010b). It is a simple web interface for launching complex applications on grid. Gridcom splits input data into intervals; generates and submits as many jobs as needed. It can scatter parametric jobs into simple jobs or resubmit aborted jobs. Finally, Gridcom collects, merges and visualizes the computed results. All these functions are performed automatically, including transparent upload of large files to storage elements. Gridcom launches specially developed Gridcom applications. Once created, an application can be launched many times with different input data. Generally it takes some hours to adapt grid application for Gridcom. Web form for grillage optimization application is shown in Fig 3.

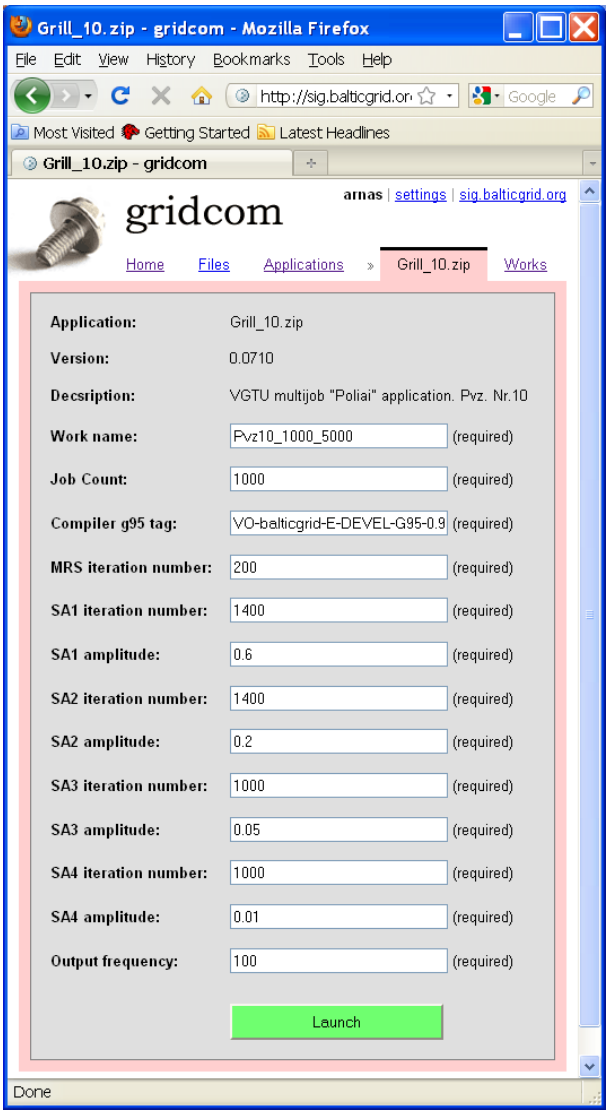

Fig 3. Web form for grillage optimization application

GRID computations have some limitations: all the numerical tests on the nodes of GRID must be accomplished in a real time (usually, in a few hours). Since some computers in the grid may be incapable, it is not possible to explore long lifespan of each run, however, a large number of the nodes can be taken. Here we render results (Šešok *et al.* 2010b) of two contrary strategies of grillage optimization on the GRIDs that exploit all capabilities of our GRID: first, large number of independent runs of SA with a short lifespan of one run, and second, less number of independent runs of SA with a maximum possible lifespan.

For the first strategy, as in case of solution on a single PC, we choose the total number of iterations N = 5000, but 1000 independent runs. For the second strategy, 100 independent runs were performed with 100,000 iterations each. Other parameters used in the solution are given in (Šešok *et al.* 2010a, 2010b).

Obtained results show, the 35 times larger number of numerical tests as compared to the calculations on a single PC allows improving solution from 463.34 to the value 449.93, i.e. for only ~4%. The second strategy gave

the best solution of 433.42, or improving of an error by ~8 %.

Cluster computing. We used the PC cluster VILKAS (Rocks Cluster Distribution v 5.0, CentOS release 5, x86_64) at Vilnius Gediminas Technical University. The cluster consisted of 18 PCs connected by Gigabit Ethernet (D-Link DGS 1224T Gigabit Smart Switch, 24-Ports 10/100/1000Mbps Base-T Module). Hardware characteristics of the PC are listed below: Intel® Core2Quad Q6600 2.40GHz CPU (2x4MB L2 cache and bus frequency equal 1067 MHz), 2x2GB DDR2 800 RAM, 300GB HDD (SATA II Extensions and 16 MB cache), Gigabit Ethernet NIC. 340 Gflop/s performance was measured running HPL benchmark. 40 VILKAS' processors were employed in computations; each of them solved the task once. The total CPU time was about 50 hours. The average results of 40 independent samples for each of the cluster computers are in Table 6 (Šešok *et al.* 2010a).

Table 6. Optimization of 55-pile grillage: cluster, 1,000,000 iterations

Sample	Value		Sample	Value
1	426.88		**21**	467.01
2	388.64		**22**	423.25
3	421.98		**23**	463.05
4	455.17		**24**	468.59
5	429.52		**25**	480.05
6	464.72		**26**	418.57
7	408.67		**27**	423.29
8	425.45		**28**	462.71
9	440.74		**29**	420.65
10	461.86		**30**	441.54
11	417.97		**31**	455.12
12	488.54		**32**	417.27
13	458.99		**33**	403.18
14	456.22		**34**	416.97
15	458.31		**35**	457.97
16	457.55		**36**	385.15
17	462.85		**37**	430.81
18	417.69		**38**	460.69
19	397.27		**39**	460.87
20	**379.35**		**40**	461.78

The best result **(**379.35) differs from global minimum (349.05) by 9 %. Thus, this error is better than the theoretical estimation of error for the number of iterations performed. From the engineering point of view, this is an acceptable error. Fig 4 shows the graph of this decision. The average results of 40 samples are 438.37.

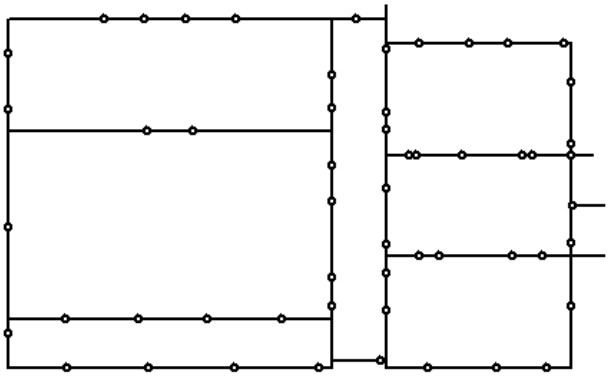

Fig 4. Best decision (55 piles, 10^6 iterations)

Conclusions

From several global optimization algorithms belonging to the stochastic algorithms (genetic algorithms, simulated annealing, modified random search, and also the simplex since it is launched from a random initial start) and to the gradient ones (BFGS, NEWUOA), the simulated annealing gives the most promising results in optimization of grillages.

Computer hardware that is common to a typical civil engineering design bureaus and a reasonable computation time for engineering practice do not allow solution of the problem of pile placement to the very end with any global optimization algorithm applied. However, first, the fact that ideal solution usually is not required for the engineering purposes, and second, the increasing accessibility and popularity of the distributed computing in contemporary engineering practice makes it possible solving global optimization problems possessing to 30–40 design parameters.

The larger optimization problem with 55 decision variables was solved only to a discrepancy between obtained result and ideal one of 9% using a cluster of PC and 50 hours of computations.

Acknowledgements

The computational resources provided by the BalticGrid-II infrastructure (FP7-INFRA-2007-1.2.3: e-Science Grid infrastructures contract No 223807) are gratefully acknowledged. This work is supported by project GridTechno (Lithuanian State Science and Studies Foundation, B-07043).

This work is also supported by The Lithuanian State Science and Studies Foundation within the project on S-20/2009 "Use of stochastic global optimization methods in engineering".

References

Belevičius, R.; Valentinavičius, S. 2001. Optimisation of grillage-type foundations, in *Proc. of the 2nd European ECCOMAS and IACM Conference "Solids, Structures and Coupled Problems Problems in Engineering"*. Cracow, Poland 26–29 June, 2001. CD-ROM.

Belevičius, R.; Valentinavičius, S.; Michnevič, E. 2002. Multilevel optimization of grillages, *Journal of Civil Engineering and Management* 8(2): 98–103.

Belevičius, R.; Šešok, D. 2008. E. Global optimization of grillages using genetic algorithms, *Mechanika* 74(6): 38–44.

Belevičius, R.; Ivanikovas, S.; Šešok, D.; Valentinavičius, S.; Žilinskas, J. 2010. Optimal placement of piles in real grillages: experimental comparison of optimization algorithms, *Engineering Optimization* (Under Review).

Chamoret, D.; Qiu, K.; Labed, N.; Domaszewski, M. 2008. Optimization of Truss and Grillage Structures by a Non-Deterministic Method. In B.H.V. Topping, M. Papadrakakis (Editors), in *"Proceedings of the Ninth International Conference on Computational Structures Technology", Civil-Comp Press, Stirlingshire, UK, Paper 51, 2008.* doi:10.4203/ccp.88.51

Chan, C. M.; Zhang, L. M.; Jenny, T. M. 2009. Optimization of pile groups using hybrid genetic algorithms, *Journal of Geotechnical and Geoenvironmental Engineering* 135(4): 497–505.

doi:10.1061/(ASCE)1090-0241(2009)135:4(497)

Čiegis, R.; Baravykaitė, M.; Belevičius, R. 2006. Parallel global optimization of foundation schemes in civil engineering. Applied Parallel Computing: *7th International Conference, PARA 2004, Lyngby, Denmark 20-23 June, 2004:* revised selected papers. *Lecture Notes in Computer Science*, 3732: 305–313. Springer.

Goldberg, D. E. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning.* New York: Addison-Wesley. 412 p.

Kim, Y.; Gotoh, K.; Kim, K. S. Toyosada, M. 2005. Optimum Grillage Structure Design under a Worst Point Load using Real-coded Micro-Genetic Algorithm, in *Proc. of the Fifteenth International Offshore and Polar Engineering Conference*, Seoul, Korea, June 19-24, 2005. 730–735.

Kirkpatrick, S.; Gelatt, C. D. J.; Vecchi, M. P. 1983. Optimization by simulated annealing, *Science* 220(4598): 671–680. doi:10.1126/science.220.4598.671

Mockus, J. 1967. *Multimodal problems in engineering design.* Moscow: Nauka.

Powell, M. J. D. 2006, The NEWUOA software for unconstrained optimization without derivatives. Di Pillo, G.; Roma, M. (Eds.) *Large-Scale Nonlinear Optimization.* Vol. 83 of *Nonconvex Optimization and its Applications*, Springer, 255–296.

Press, W. H. 1992. *Numerical Recipes in Fortran 77: The Art of Scientific Computing. Second Edition*. Cambridge University Press.

Rozvany, G. I. N. 1997. Exact optimal layout of grillages for partially upward and partially downward loading, *Structural and Multidisciplinary Optimization* 13(4): 267–270.

Šešok, D.; Belevičius, R 2008. Global optimization of truss systems using a modified genetic algorithm. *Journal of Civil Engineering and Management*, 14(3): 147–154. doi:10.3846/1392-3730.2008.14.10

Šešok, D.; Mockus, J.; Belevičius, R.; Kačeniauskas, A. 2010a. Global optimization of grillages using simulated annealing and high performance computing, *Journal of Civil Engineering and Management* 16(1): 95–101. doi:10.3846/jcem.2010.09

Šešok, D.; Mockus, J.; Belevičius, R.; Kačeniauskas, A. 2010b. Application of GRID computing for optimization of grillages, *Mechanika* (accepted).

Zienkiewicz, O. C., Taylor, R. L. 2005. *The Finite Element Method for Solid and Structural Mechanics.* London: Elsevier Butterworth-Heinemann, 1–632 p.