ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 2012
Seria: TRANSPORT z. 76 Nr kol. 1865

Marián HANDRIK[1], Milan VAŠKO[1], Peter KOPAS[1]

PARALLEL AND DISTRIBUTED IMPLEMENTATION OF OPTIMIZATION ALGORITHMS IN FE ANALYSES

Summary. The aim of this paper is analysis of optimization algorithms in terms of their possible solutions in parallelization and distributed computing systems. Main goal is using of evolutionary algorithms and implementation of parallel algorithms. As the software platform for application of distributed optimization algorithms is using software package BOINC. For evaluation of the objective function is used FEM program ADINA.

Keywords. FE analysis, optimization, parallel and distributed computing, BOINC, grid computing, evolutionary algorithms.

IMPLEMENTACJA OPTYMALIZACYJNYCH ALGORYTMÓW RÓWNOLEGŁYCH I ROZPROSZONYCH W ANALIZIE MES

Streszczenie. Artykuł analizuje algorytmy optymalizacyjne pod kątem ich możliwości obliczeń równoległych oraz rozproszonych systemów obliczeniowych. Ukierunkowany jest przede wszystkim na algorytmy ewolucyjne oraz ich implementację równoległą. Jako platforma softwarowa do zastosowania rozproszonego systemu obliczeniowego algorytmu zostało zastosowane oprogramowanie pośredniczące BOINC. W celu oceny funkcji docelowej został zastosowany w MES program ADINA.

Słowa kluczowe. Analiza MES, optymalizacja, obliczenia równoległe i rozproszone, BOINC, rozproszony model obliczeniowy, algorytmy ewolucyjne.

1. INTRODUCTION

Solving of the optimization problem is generally based on the repeated evaluation of the objective function. Obtained solutions are comparing and better solution is selecting. Evaluation of the objective function represents FE analysis of the issue in the optimization problems in continuum mechanics [3, 4, 6].

Therefore it is necessary to implement several types of analysis based on FEM in solving of the complex optimization problems (stress analysis to determine the state of stress distribution, modal–frequency analysis to determine the natural frequencies, loss of stability analysis, etc.).

[1] Department of Applied Mechanics, Faculty of Mechanical Engineering, University of Žilina, Žilina, Slovak Republic,
e-mail: marian.handrik@fstroj.uniza.sk, milan.vasko@fstroj.uniza.sk, peter.kopas@fstroj.uniza.sk

One evaluation of the objective function often represents a solution with total duration of several tens or hundreds of minutes. To obtain the optimal solution is required tens to hundreds of thousands of objective function evaluations, depending on the used optimization method and the number of design variables [2]. The solution of such problems is unrealistic using conventional optimization algorithms. This solution could take several years.

For the purpose of more efficiently solve the optimization problem is necessary to modify optimization algorithms. The ADINA software package will be used for evaluating the objective function value in the implementation of parallel optimization algorithms. It can be define parameters and create a parametric model in the input file. This enables easy change of design variables in optimization process.

Computational Systems

Using the parallel solution of linear equations in the FEM analysis is a fundamental possibility of increasing the speed of the optimization problem solution. The two fundamental architectures of computational systems can be used in the parallel solving the system of equations:

- SMP technology (Symmetric Multi Processing) – this means multiprocessor and multicore computers,
- DMP technology (Distributed Multi Processing) – this is a group of computers that are connected to a computer network and computers work together to solve problems. Mutual communication between computing nodes is often realized using MPI technology, etc.

Iterative solvers are used for solving systems of linear equations in large-scale problems. Parallel implementation of iterative solvers is effective only for a relatively small number of compute cores that are used for the problem solution. Increasing of the computing power is sometimes possible by using graphic accelerators, i.e. special graphics cards designed for the realization of calculations directly on these cards.

In the case of more required and solving tasks and provided adequate computing resources power to implement they the following calculation procedure can be used:

- Serial problem solving using the whole computing power – in this case the nodes are often only partially used.
- Parallel solutions of multiple tasks on the smaller number of computational cores – individual computing nodes are used more efficiently.

Practical tests have shown that in case of sufficient capacity of computing resources the parallel solution in terms of duration of solving across a series of tasks appears more efficiently. Alternatively, increasing the speed of solving the optimization problem is the use of the above properties and modification of the optimization algorithms to the parallel implementation.

2. THE OPTIMIZATION ALGORITHMS

Optimization methods can be divided into analytical and iterative [5]. If in solving complex optimization problems of continuum mechanics to evaluate the objective function is using commercial FEM program, it is possible the iterative methods use only in the optimizing process.

Iterative optimization algorithms can be divided into three basic groups. The division factor is the relationship of the algorithm to process the evaluation of the objective function:

- The direct methods – evaluates the value of the objective function.

- The first order methods – there needs to be evaluated function gradient and to evaluate the gradient is used numerical differentiation.
- The second order methods – should be evaluated Hess matrix. Hess matrix is derived numerically similar to the first order methods. The numerical experiments showed that the numerical differentiation of second order in relation with evaluating the objective function and using the FEM program compiles Hess matrix with a large error. These methods then become non-convergence.

The group of direct methods can be divided into two subgroups – methods of random search and evolutionary algorithms [7]. Their parallelization is simple – it is sufficient to implement a parallel evaluation of objective function, i.e. multiple FEM analyses solution work simultaneously. A subset of deterministic methods after their parallelization is transformed into the first subset or the gradient methods. Parallelization of the first order methods is based on the process of numerical differentiation parallelization.

Based on the possibility of parallelization the iterative optimization algorithms can be divided into the following groups:

Algorithms that cannot be parallelize

This group can include algorithms where parallelization is possible only in a few steps of the algorithm. This group includes the Simplex algorithm, in which is only possible to parallelize the compilation of the initial simplex.

Algorithms with the interaction

This group consists of algorithms in which needs to be completed one step of algorithm to algorithm can continued in the calculation. Typical representatives of these algorithms are the first and second order algorithms – gradient methods.

With these methods it is possible to parallelize the gradient or Hess matrix compilation while algorithm must wait for their establishment. Therefore limiting factor for running the program is waiting for compilation the last element of the gradient or Hess matrix.

Suitable computational resources for this type of algorithms are those where each computing node has the same computing power. Then the powerful computational nodes do not have to wait for less powerful computing nodes.

Algorithms without the interaction

Algorithms included in this group have to solve a high degree of tolerance to time and return the objective function solution. Typical representatives of this group of algorithms are evolutionary algorithms. The descendants group with appropriate genetic variations is generating in the vicinity of the parent. The value of the objective function will be subsequently evaluated for this group of descendants. Member with the best objective function value becomes the new parent.

The following modification can be used for parallelization of evolutionary algorithms: Packet of descendants is created into which shall be supplemented continuously descendants generating on the base of current parents. These descendants are waiting to evaluate the objective function. The next step is parallel evaluation of descendants waiting in the packet and comparing the value of the objective function with a parent. If a descendant has a better objective function value than the current parent this child becomes the new parent.

In the packet of descendants may be found simultaneously descendants from the different parent generations but better parent can be found in the previous generation as is the current parent. Parallel evaluation of the descendant's objective function is mutually independent and sufficient condition the reasonable length of time for return the objective function.

This condition allows into simultaneous evaluation include the computing resources with significantly different computing power (Fig. 1).

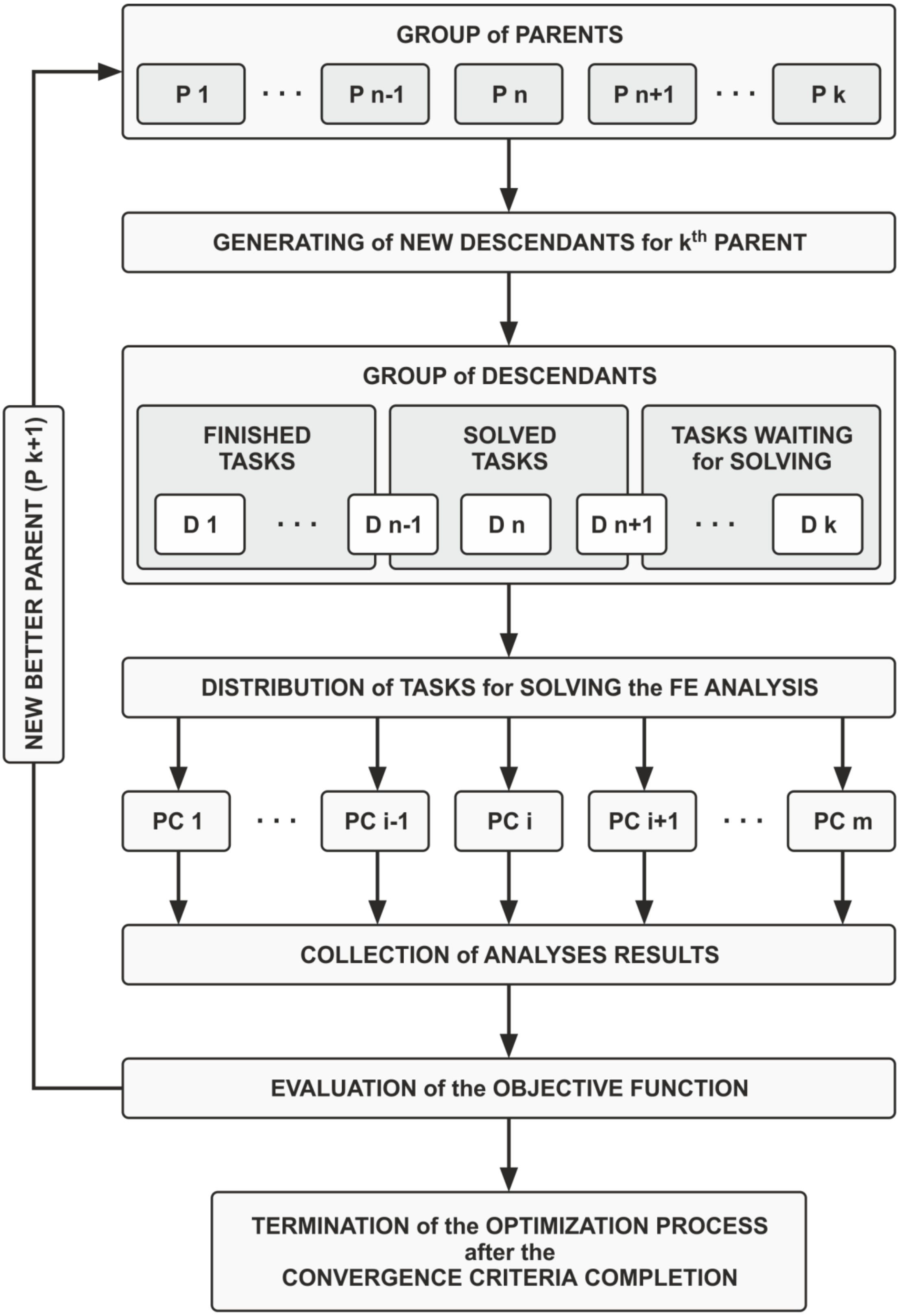

Fig. 1. Simplified scheme of the optimization process for the grid infrastructure
Rys. 1. Uproszczony schemat systemu optymalizującego dla rozproszonego systemu obliczeniowego.

3. EVOLUTIONARY ALGORITHMS AND GRID COMPUTING

Gradient and evolutionary algorithms are significant in terms of optimization algorithms parallelization with the use of a commercial FEM program for evaluating the value of objective function. Problems of continuum mechanics cannot be considered as task with the one global extreme. More local extremes are often found in the solution area.

The use of gradient methods leads to finding the local extreme, depending on the position of the starting point or starting vector. Evolutionary algorithms are suitable method for finding global extreme of the objective function in relation to the optimization process parallelizing.

The term grid computing represents group of different computers that work together to solve complex computing tasks. Complex task is to find the optimal solution of the problem in terms of the optimization process. Partial task is to determine the value of the objective function for the vector of design variables. In evolutionary algorithms it represents determination the value of objective function for a specific descendant. There are a number of software platforms for the implementation of grid computing today.

BOINC (Berkeley Open Infrastructure for Network Computing) has been used to implement a distributed computing model. It's open source system for distributed computing. The robustness, flexibility and universality of the system is verified on a number of world famous projects in the area of grid computing (SETI, Milkyway, LHC, Einstein, etc.). The order of one million computers is participating in the solving the tasks [1].

Computing system consists of server and client part. The server part is designed to ensure the following functions:

- generate work units (work unit generator),
- insertion of work units into a set of tasks to send (feeder),
- identification of client's request and schedule its compliance (scheduler),
- client requirements completion – download work units and upload results (CGI scripts in Apache),
- comparison of the results for individual work units (validator),
- processing results after their validation (assimilator),
- provision of communication and reporting information about the process of the work units solution (transitioner),
- release disk space by deleting of solved work units (file deleter).

The client part is designed to ensure the following functions:

- download the main program for work unit solving, monitoring updates and ensure downloading a newer version,
- enable logging to the project or multiple projects,
- allocation of free system resources for their use,
- sending requests for the allocation of work units and their download on the client computer,
- send the information about status and results of solution.

A computer running Debian Linux 6.0 has been used to implement a distributed computing system. BOINC was installed directly into the system without using virtualization.

To adjust the BOINC system for needs of optimization model implementation was necessary to perform the following changes:

- change the general system settings,
- adjust the work units generator,

- create a client application,
- adjust the processing of the results, modifying the assimilator.

General settings change consists of changing the approach of the feeder to the prepared work units. The strategy of work units' choices was changed for insertion in the system FIFO (First In First Out). Work units, i.e. individual descendants are inserting from the oldest to newest for solving in the evolutionary strategy. Other changes consist in reducing the feeder tray for ten work units. It is sufficient for the number of 22 client computers.

At the editing of work units generator it is based on the assumption that its task is generate new descendants with a random genetic mutation in the best parent's neighborhood. The generator monitors the number of prepared work units waiting to be processed. The best parent parameters are loaded and runs generate a new group of descendants if the number of prepared work units decreases under the specific bound.

It is monitoring before generating of descendants if in terms of the solution convergence the generating is needed. The creation of work unit consists of generating random genetic mutations, preparation of input files for the ADINA preprocessor, preparing files for the ADINA postprocesing and compression of all files to the "7zip" archive.

The advantage is that at any modification in the optimization task there is always necessary to transfer only one file. The optimization problem change is therefore realized by the change of the input files for work units generator on the server side. Therefore is not necessary to change the work unit's generator.

The role of the client application is to ensure the following functions:

- extract the files from the work unit archive,
- run ADINA preprocessing and then solving tasks in ADINA,
- write selected results into files by running ADINA postprocesing,
- archiving initial and executive programs by "7zip" into a single file for transmission to the server.

The assimilator modification is in preparing them to execution the following steps:

- The result of the analysis solved by FEM in the program ADINA can contain multiple numeric data recorded in different files [8]. The assimilator compiles objective function value from the partial results and then determines if the evaluated descendant becomes new parent.
- If it becomes the new parent then into the relevant files are written new parent parameters and its objective function value.
- In the evaluation of the each work unit convergence criterion is checked and in the case of its compliance gives the instruction to the termination of generating work units.
- The condition was used in the choice criteria of convergence if enough new descendants were generated without finding a new better parent.

It is necessary to make appropriate adjustments in the assimilator when is changing the form of the target function.

4. NUMERICAL EXPERIMENTS

The implementation of a simple evolutionary algorithm for optimizing with the use of the distribution system for grid computing BOINC, including its debugging runs two weeks. The server part tests of the computing system were done on a computer configuration

(8xAMD OPERTON 8378, 128 GB RAM, 8x1 TB HDD, software raid 5) and showed the following results:

- The generation speed is about 500 work units per minute for the one work unit generator.
- Processing speed of the completely counted work units by validator and assimilator is about 2000 per minute for one work unit generator.
- FEM analysis on the client computer for the realization of shell construction optimization with a total of about 2 million equations takes approximately 9 minutes. One to two work units were located on the client computer. FEM analysis was performed at the one work unit and another work unit is waiting for FEM analysis or is sending the results. FEM analysis results are usually returned for processing within one minute after its finalization.
- The results are processed on the server for realized work unit in one minute of its recovery from the server.

5. CONCLUSIONS

Use of the BOINC as a control system for grid computing allows very fast response of the system at the appropriate setting of its parameters (the short time between the end of the FEM analysis and processing of the results in the assimilator for the work units generator). This allows the implementation of evolutionary algorithms that require a higher level of feedback between the generated descendants and actual parents.

A small claims rate on system resources, parallel realization of the work units generation and parallel processing of the results enables involving the order of several thousand to tens of thousands of PCs in FEM analyses necessary for solving the optimization problem.

This robust computing infrastructure allows the realization of optimization problems for finding the global extreme in continuum mechanics problems with hundreds of design variables in relatively short time.

This work has been supported by VEGA grant No. 1/1089/11.

This contribution is the result of the project implementation: Development of optimum technology for the analysis of limit states of structural elements in contact, ITMS code 26220220118, supported by the Research & Development Operational Programme funded by the ERDF.

We support research activities in Slovakia /
The project is co-financed by the European Union.

Bibliography

1. http://boinc.berkeley.edu/
2. Grega R., Kaššay P., Krajňák J.: *Impact of Pneumatic Flexible Coupling on Effective Vibration Value within the Mechanical Systems*. Transactions of the Universities of Košice, Košice, No. 3, 2009, ISSN 1335-2334, p. 5–8.
3. Kaššay P., Homišin J., Grega R., Krajňák J.: *Comparation of selected pneumatic flexible shaft couplings*. In: Zeszyty naukowe Politechniki Śląskiej. Vol. 73 (2011), p. 41-48. ISSN 0209-3324.
4. Sága M., Vaško M.: *Algorithmization of Interval Structural Analysis*. In: Journal of Engineering, Annals of Faculty of Engineering Hunedoara, Tome VI, 2008, ISSN 1584-2665, p. 149–160.
5. Sága M., Vaško M., Kocúr R., Toth Ľ., Kohár R.: *Aplikácia optimalizačných algoritmov v mechanike telies*. VTS pri ŽU v Žiline, 2006. ISBN 80–969165–9–9.
6. Vaško A.: *Image analysis in materials engineering*. In: Konferencje, Poland, 2007, No. 61, p. 667–670, ISSN 1234-9895.
7. Vaško M.: *Application of Genetic Algorithms for Solving of Mechanical Systems with Uncertain Parameters*. PhD. thesis, Žilina, 2007.
8. Vaško M., Sága M., Handrik M.: *Comparison Study of the Computational Methods for Eigenvalues IFE Analysis*. In Applied and Computational Mechanics. ISSN 1802–680X, 2008, vol. 2, no. 1, p. 157–166.