

Numerical Optimization using Stochastic Search Algorithms and Finite Element Function Evaluations in a Parallel Environment

Uwe Pahner, Kay Hameyer and Ronnie Belmans
KATHOLIEKE UNIVERSITEIT LEUVEN, DEPT. E.E. (ESAT), DIV. ELEN
Kardinaal Mercielaan 94, B-3001 Leuven, BELGIUM

Abstract - Stochastic optimization algorithms require substantially more function evaluations compared to gradient methods. This results in a long overall computation time. However, the stochastic algorithms feature unmatched simplicity in setting up the optimization problem. A reduction of the computation time can be achieved with the parallel implementation of the optimization procedure on a network of heterogeneous computers. Here, a parallel implementation of the Evolution Strategy is presented. Several load balancing and scheduling schemes as well as the necessary adaptations to the finite element (FE) data structures are discussed.

INTRODUCTION

An optimum design is defined as the best possible solution for a given application. All design variables are determined simultaneously to satisfy a set of constraints and optimize a set of objectives, represented by a quality function. It is widely accepted, that the main advantages of stochastic search algorithms (Evolution Strategy, Simulated Annealing and Genetic Algorithms) is their robustness, ease of use and general application range [1]. Set-up and testing of a new optimization problem (quality function, implementation of constraints) requires far less effort compared to deterministic algorithms. A drawback is their huge number of necessary function evaluations, increasing the computation time. The rapid enhancement of computer power and the availability of new computer architectures (parallel machines) help to reduce the overall computation time. An implementation using *PVM* (*Parallel Virtual Machine*) is described here [2]. Combining stochastic optimization algorithms and field simulation techniques into an optimization environment allows the creation of a user friendly design tool [3]. Necessary adaptations to the finite element data structures and the optimization algorithm will be described, before the parallel implementation of the optimization procedure is discussed in detail.

PARAMETRIC FINITE ELEMENT MODELS

A stand-alone, fully interactive and parametrized 2D FE-package, embedding different solvers and post-processor tools, has been developed. The 2D pre-processor includes all features that are expected from a classical FE pre-processor, but additionally provides all tools to set-up, test and control optimization tasks, such as:

- definition and test of analysis procedures including different types of solvers and post-processor routines,
- definition of constraints checks and normalizing factors for the design variables,
- storage of the model and the procedure in a symbolic format,
- automatic preparation of optimization tasks for a parallel environment using *PVM*.

GENERAL PURPOSE OPTIMIZER

A general purpose optimizer has been developed, inheriting different stochastic optimization algorithms, such as Evolution Strategy, Simulated Annealing and Genetic Algorithms. The optimizer can operate in an external mode. Therefore, it has not to be adapted to different types of analysis procedure (numerical, analytical) [3]. Typical for stochastic optimization algorithms is the generation of independent sets of parameters within one iteration. This allows a parallelization of the optimization process.

PARALLEL IMPLEMENTATION OF THE OPTIMIZATION PROCESS

The parallel environment consists of a variable number of computers in one network. A master process on one of the machines controls slave processes on the other nodes of the network. Fig. 1 outlines the implemented algorithm. All beforehand defined procedures for the FE analysis together with the parametric description of the model are distributed amongst the local computers of the parallel environment at start-up time. The symbolic descriptions reside local on all machines during the whole optimization process. To reduce

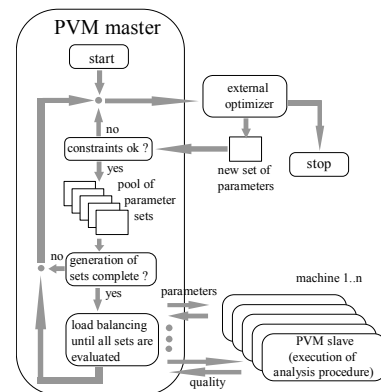


Fig. 1: Parallel implementation of the stochastic search algorithm using *PVM*

the influence of the network load, the data exchange between the master and the slave processes (during the optimization) is limited to the updated values of the design variables, the value of the quality function and the status of the single function evaluation. The parameter pool is filled with n -sets of parameters, representing the n -design candidates to be evaluated in one iteration during the optimization. A new set can only be generated after all qualities of the present iteration are evaluated. These processes have to be spawn onto the different machines. Towards the end of each iteration, a situation may occur, were a slower machine could start a function evaluation, which could take longer than the faster machines would require to evaluate the remaining parameter sets in the pool. This would unnecessarily increase the elapsed optimization time. Also, the influence of third party loads (other network users) can not be neglected in a realistic networking situation (Fig. 2).

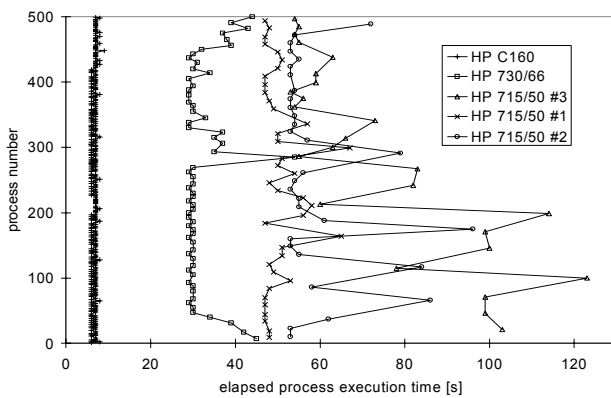


Fig. 2: Influence of third party load on the elapsed function evaluation time over 20 iterations (25 identical function evaluations per iteration)

Fig. 3 illustrates the effect of the different machine load levels. Slower machines have longer idle periods (up to 50% of the overall optimization time), because faster machines are preferred towards the end of each iteration. However, the overall runtime is reduced to 67% of the time the fastest machine alone would need for this task.

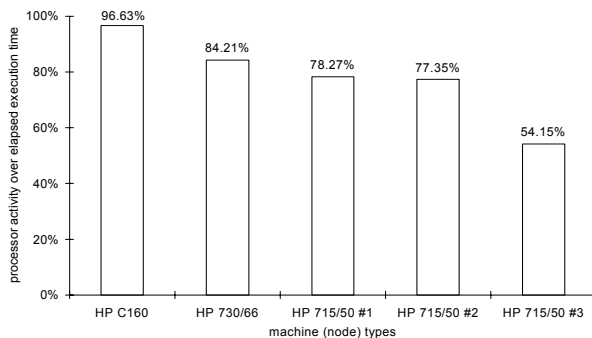


Fig. 3: Effective function evaluation time per node (in percent of the total computation time needed for the 500 identical evaluations shown in Fig. 2)

The aim of the parallelization of the optimization procedure is to reduce the elapsed optimization time to a minimum. Situations as shown Fig. 3 are not acceptable. The elapsed computation time for one function evaluation not only varies due to third party load, but different parameter sets also lead to different function evaluation times. Dynamic load balancing and scheduling are necessary to make optimum use of the parallel environment. Apart from scheduling algorithms based on the monitoring of the evaluation times, most stochastic optimization algorithms offer another possibility to reduce the optimization time: The number of function evaluations per iteration may be varied to a small extent. This allows long lasting function evaluations towards the end of an iteration to be transferred to the next iteration. The optimum use of all machines in the environment can now be assured. Different scheduling schemes will be compared in the full paper. Tests have shown that in a set-up with identical machines (5 HP 715, no third party load), the down-scaling is almost linearly depending on the number of machines included. All machines have idle times of less than 3% and the overall computation time is reduced to 20.5% of a single machine execution. This can only be achieved due to the minimized data exchange with the master process. The exchanged data packages are of the size of some bytes, whereas the FE data structures can reach several Mbytes during the evaluation of the quality function.

CONCLUSION

A parallel implementation for the optimization of electromagnetic devices has been developed. It is cost effective, due to the efficient use of existing hardware. The emphasis is put on the development and comparison of appropriate load balancing and scheduling schemes, making optimum use of the features of the stochastic search methods in combination with the finite element analysis.

ACKNOWLEDGMENT

The authors are grateful to the Belgian "Fonds voor Wetenschappelijk Onderzoek Vlaanderen" for its financial support of this work and the Belgian Ministry of Scientific Research for granting the IUAP No. P4/20 on Coupled Problems in Electromagnetic Systems. The research Council of the K.U.Leuven supports the basic numerical research.

REFERENCES

- [1] A. Gottvald, K. Preis, Ch. Mangele and A. Savini, "Global optimization methods for computational electromagnetics," *IEEE Transactions on Magnetics*, vol. 28, no. 2, 1992, pp. 1537-1540.
- [2] A. Geist, A. Beguelin, J. Dongorra, W. Jiang, R. Manchek and V. Sunderam, *PVM: Parallel Virtual Machine, A Users' Guide and Tutorial for Networked Parallel Computing*, The MIT Press, Cambridge, Massachusetts, London, England, 1994.
- [3] U. Pahner, R. De Weerd, K. Hameyer, R. Belmans: "A parametric environment for the optimization of electromagnetic devices" 3rd international workshop on Electric & Magnetic fields, Liège, Belgium, May 6-9, 1996, pp.177-182.