ECG Simulator Tuning: A Parallel Multiobjective Optimization Approach

Bogdan Filipič[1], Matjaž Depolli[1], Jernej Zupančič[1], Jan Gmys[2,3], Maxime Gobert[3], Nouredine Melab[2] and Daniel Tuyttens[3]

[1] Jožef Stefan Institute, Ljubljana, Slovenia
{bogdan.filipic,matjaz.depolli,jernej.zupancic}@ijs.si
[2] Inria Lille - Nord Europe, CNRS/CRIStAL, Université Lille 1, France
nouredine.melab@univ-lille1.fr
[3] Mathematics and Operations Research Department, University of Mons, Belgium
{jan.gmys,maxime.gobert,daniel.tuyttens}@umons.ac.be

1 Introduction

Devoted to enhancing multiobjective optimization through parallel computation and surrogate modeling, the Horizon 2020 project SYNERGY [1] uses the electrocardiogram (ECG) simulator tuning as a benchmark problem to exploit the synergistic effects of parallelization and surrogates in solving multiobjective optimization problems. Here we report on the initial work and findings on this task by the project partners Jozef Stefan Institute (JSI) and University Lille 1. We present the background on ECG and its simulation, the optimization task and the initial experiments and results, and conclude with a summary and directions for further work.

2 ECG and its simulation

The standard 12-lead ECG has been a diagnostic tool for over 70 years. It is a non-invasive procedure used to monitor the heart's electrical activity that arises from rhythmical contractions of the heart muscle (myocardium) pumping the blood throughout the body. Electrical currents are generated in myocardium as side effects of the contractions. Since the human body is electrically conductive, electrical currents can be detected on the body surface where they are measured by an ECG machine. ECG readings can be observed for deviations of ECG shape from the typical or normal ECG shape, as some diseases and health conditions cause observable and well known symptoms in the ECG.

Although the normal ECG shape and some typical defects are well known, the transfer function that maps the ECG measured on the body surface to individual cells of myocardium is not known. Gathering additional knowledge on the transfer function would help improve ECG based diagnostics and enable better prediction of health condition, based on the ECG reading.

One of the basic tools for studying the transfer function and for unveiling additional knowledge on heart activity is a computer simulator of the human heart. In this work we use the simulator [3, 4] that was utilized to gain new insights into possible shapes of action potentials (APs) in myocardium. APs represent voltage as a function of time for an individual cell. Voltage is measured as the potential difference of cell exterior against the cell interior. APs of human heart cells can be modeled as a system of non-linear time-dependent differential equations [5], which is quite time consuming. Therefore it was approximated by a combination of exponential functions [6] to make the computer simulation of a large number of cells feasible. The combination of exponential functions was further refined [4], after it was discovered it contains a physiologically unrealistic symmetry. The function $AP(t)$ is parameterized with nine parameters $k_0, k_1, ..., k_8$ and can now be written as:

$$\begin{aligned} A(t) &= \tfrac{1}{1+e^{-k_1 t}} \\ B(t) &= k_2((1-k_3)e^{-k_4 t} + k_3)e^{-k_5 t} \\ C(t) &= 1 - (1 + e^{-k_7(t-k_8)+\ln(2^{\frac{k_7}{k_6}}-1)})^{\frac{-k_6}{k_7}} \\ AP(t) &= k_0 + A(t) \cdot B(t) \cdot C(t) \end{aligned} \tag{1}$$

In this work we widen the functionality the simulator to include the simulation of the first 100 ms of the heart beat, which was previously not done. This part of the heart beat is more complex than the rest since it comprises very quick changes (large derivatives in APs and in the resulting ECG) and thus requires small simulation time steps and a detailed heart model.

3 The optimization task

The ECG simulator operates in a closed loop with the optimization process in an effort to fine-tune the simulator to produce realistic ECGs. Out of nine AP function parameters, two (k_0, k_2) have predefined values, while the remaining seven are subject to optimization. As three layers of myocardium cells are considered in the model, the total number of optimization variables is $3 \cdot 7 = 21$.

The simulator output is assessed by comparing the output of the simulation, i.e., two simulated ECGs at different positions on the body surface, to measured ECGs at the same locations on the body surface. The objectives are to maximize the Pearson correlation coefficients between the measured and the simulated ECG signals for the two positions. The simulator is unable to simulate the actual ECG amplitudes as its focus is on the signal shape. Therefore, the objectives were selected to ensure that the difference in ECG amplitudes is not considered, while the features of the simulated ECGs should match the measured ECG features in time.

4 Initial experiments and results

The multiobjective optimization algorithm used to tune the ECG simulator was Asynchronous Master-Slave Differential Evolution for Multiobjective Optimization (AMS-DEMO) [7], which is an extension of the DEMO algorithm [8, 9]. AMS-DEMO was designed for solving computationally expensive problems on homogeneous and heterogeneous parallel computer architectures. It assumes all objectives are to be minimized. For this reason, the original optimization requirement to maximize the Pearson correlation coefficient for each of the two pairs of ECG signals was reformulated to minimize the value of 1−Pearson correlation coefficient for each pair.

The purpose of the initial experiments was to assess the performance of AMS-DEMO on the ECG simulator tuning problem and study how it can benefit from parallelization. In all experiments the following algorithm parameter settings were applied: AMS-DEMO based on DEMO/parent and using the DE/rand/1/bin scheme, crossover probability 0.5, scaling factor 0.5, populations size 100, and the number of generations 100. No tuning of the algorithm parameters was carried out at this stage. The experiments were performed on JSI and Lille 1 clusters.

4.1 Parallelization on JSI cluster

The NSC cluster at JSI comprises 1984 compute cores, 9216 GB RAM and 16 graphical processing units (GPUs). It is accessible using the NorduGrid ARC client, and jobs are submitted using the SLURM job scheduler. The optimization source code consisting of the AMS-DEMO and the ECG simulator [2] was augmented for running on this infrastructure. Each job (AMS-DEMO run) was specified to use Open MPI 2.0.1, 64 computing cores (on several available nodes) and up to 500 MB RAM per core.

Five test runs were executed. The runs took from 1 day and 19 hours to 2 days and 20 hours, depending on the cluster load, with a solution evaluation (ECG simulator run) taking 20.6 minutes on average. The progress of the hypervolume indicator through the optimization runs is shown in Fig. 1. The reference point was set at (2.1, 2.1), since 2 is the maximum value that can be obtained in each objective. The maximum hypervolume was obtained in the last generation, achieving an average of 2.69 over all runs.

4.2 Two-level parallelization on Lille 1 cluster

To further accelerate the ECG simulator tuning process, a second level of parallelism was added at the instruction level, leveraging SIMD (single-instruction multiple-data) processing capabilities.

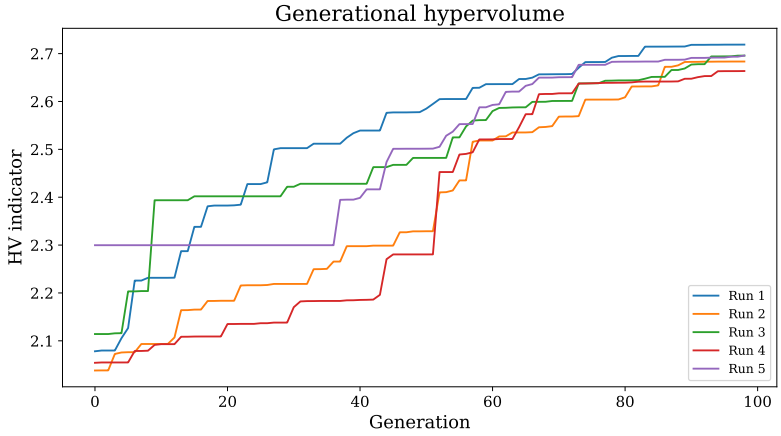

Fig. 1: Hypervolume indicator throughout the test runs

The targeted hot spot was the ECG simulator, by far the most time-consuming part of the optimization process. Indeed, most modern multi-core processors provide vector instruction sets that allow to carry out multiple instructions in one clock cycle, e.g., perform identical floating-point operations on up to four double-precision operands that reside in 256-bit wide vector registers (AVX2). In some cases, automatic vectorization can be achieved by enabling architecture-specific optimization at compile-time. However, a finer control of the vectorization mechanism is often necessary, either by inserting hints to the compiler in the source code, or by explicitly using vector intrinsics/assembly language.

Efficient vectorization of the ECG simulator is achieved by (1) a modification of the memory layout, (2) the insertion of `#pragma` preprocessor directives, and (3) an appropriate alignment of data structures in memory. As it can be seen in Table 1, the revised implementation of the ECG simulator accelerates the evaluation of solutions by an average factor of 3.2 (of a possible 4), compared to the auto-vectorization approach.

Table 1: Single ECG simulation: average over 10 evaluations of random solutions

Compiler/Libraries	Code	$T_{\text{evaluation}}$ (min)
GCC/GNU	base	9.3
ICC/Intel	base	4.2
ICC/Intel	vect	1.3

Combining the node-level parallelism of AMS-DEMO with the SIMD-vectorized ECG simulator, the optimization is completed within 48 minutes, using 320 CPU cores distributed on 20 nodes of the Lille 1 cluster (20×2 8-core Sandy Bridge E5-2670 processors, using OpenMPI 2.0.1 and Intel ICC 17.0.1). As shown in Fig. 2, this amounts to performing 210 ECG simulations per minute on average. At constant workload and up to 320 cores, the execution time of AMS-DEMO decreases almost linearly with the number of processors. The decrease by a factor greater than 2 – when increasing the core count from 16 to 32 – can be explained by the fact that the master process resides on the first host processor, slowing down the workers on this node.

5 Conclusion

We have presented the initial experiments and results in parallel multiobjective optimization of the ECG simulator that serves as a SYNERGY project benchmark problem. Concerning the solution quality, the results show that further improvements are possible both at the level of the simulator accuracy and the optimization algorithm tuning. In particular, the ECG simulator takes good advantage of vector processing units, indicating acceleration potential on coprocessors like Xeon

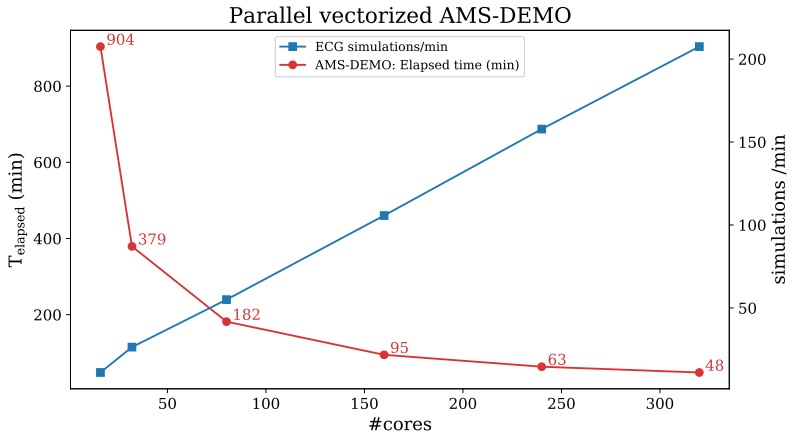

Fig. 2: Strong scaling of vectorized AMS-DEMO on a cluster composed of 108 dual-socket nodes (8-core Xeon E5-2670), using 1, 2, 5, 10, 15 and 20 nodes

Phi and GPU. Further speedup should allow a refinement of the time and space discretization in ECG simulation. The ability of performing faster and more accurate simulations can also help building better surrogate models for the fine-tuning of the ECG simulator. We plan to investigate the use of the latter in the near future.

Acknowledgment

This work is part of a project that has received funding from the *European Union's Horizon 2020 research and innovation program* under Grant Agreement No. 692286. The authors also acknowledge the financial support from the Slovenian Research Agency (research core funding No. P2-0095 and P2-0209).

References

1. SYNERGY for Smart Multi-objective Optimisation, http://synergy-twinning.eu/.
2. ECG simulator based on 3D voxel model of a human heart, https://github.com/synergy-twinning/ekgsim.
3. Depolli, M., Avbelj, V., Trobec, R. Computer-simulated alternative modes of U-wave genesis. *Journal of Cardiovascular Electrophysiology*, 19 (1): 84–89, 2008.
4. Trobec, R., Depolli, M., Avbelj, V. Simulation of ECG repolarization phase with improved model of cell action potentials. In: Fred, A., Filipe, J., Gamboa, H. (eds.), *Biomedical Engineering Systems and Technologies: International Joint Conference, BIOSTEC 2009, Porto, Portugal, Revised Selected Papers*. Communications in Computer and Information Science, vol. 52, pp. 325–332. Springer, 2010.
5. ten Tusscher, K. H. W. J., Noble, D., Noble, P. J., Panfilov, A. V. A model for human ventricular tissue. *American Journal of Physiology – Heart and Circulatory Physiology*, 286 (4): H1573–H1589, 2004.
6. Wohlfart, B. A simple model for demonstration of SST-changes in ECG. *European Heart Journal*, 8 (4): 409–416, 1987.
7. Depolli, M., Trobec, R., Filipič, B. Asynchronous master-slave parallelization of differential evolution for multiobjective optimization. *Evolutionary Computation*, 21 (2): 261–291, 2013.
8. Robič, T., Filipič, B. DEMO: Differential evolution for multiobjective optimization. In: Coello Coello, C. A., Hernández Aguirre, A., Zitzler, E. (eds.), *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization, EMO 2005, Guanajuato, Mexico*. Lecture Notes in Computer Science, vol. 3410, pp. 520–533. Springer, 2005.
9. Tušar, T., Filipič, B. Differential evolution versus genetic algorithms in multiobjective optimization. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.), *Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization, EMO 2007, Matsushima, Japan*. Lecture Notes in Computer Science, vol. 4403, pp. 257–271). Springer, 2007.