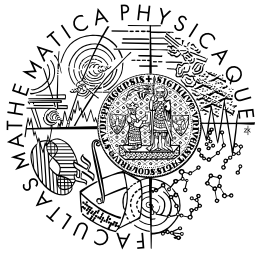

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Patrik Valkovič

GPU Parallelization of Evolutionary Algorithms

Department of Theoretical Computer Science and Mathematical Logic

Supervisor of the master thesis: Mgr. Martin Pilát, Ph.D.
Study programme: Computer Science
Study branch: Artificial Intelligence

Prague 2021

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date ....

Author's signature

I wish to thank my supervisor, Mgr. Martin Pilát, Ph.D., for his guidance, help, and mostly patience during my work on this thesis. I am thankful for his friendly attitude and valuable advices.

I would also want to express my sincere gratitude to my family and friends, without which support this work would not have been possible.

Last but not least, I would like to thank Cesnet for allowing me to use the computational power of MetaCentrum for the purpose of this thesis. I could not get presented results without their help.

Title: GPU Parallelization of Evolutionary Algorithms

Author: Patrik Valkovič

Department: Department of Theoretical Computer Science and Mathematical Logic

Supervisor: Mgr. Martin Pilát, Ph.D., Department of Theoretical Computer Science and Mathematical Logic

Abstract: Abstract.

Keywords: evolution algorithm, parallelization, GPU computing, genetic algorithms

Contents

1. Introduction

Artificial intelligence (AI) has become phenomenon of these days. Latest advances in this field achieved magnificent results and allowed creation of systems and tools, which we would not think were possible thirty years back. The biggest credit goes to the Artificial Neural Networks (ANN), which allowed this rapid and astonishing growth of the field in the last years. More precisely, the Deep Artificial Neural Networks (DANN) sparkled this process in 2011, when they started to exceed human performance on German traffic sign recognition benchmark [Cireşan et al., 2012].

Even though there were already known industrial applications of AI at the beginning of the 2000s, such as automatic check processing [Lecun et al., 1998] or speaker recognition [Heck et al., 2000], it were mainly achievements in 2011 and 2012 which acquired attention from academia and the broader community. The progress in AI is faster each day and we can observe new results almost on a daily basis. Current state-of-the-art system already exceed human performance and traditional methods in number of various tasks – like image recognition [Pham et al., 2021][Zawadzka-Gosk et al., 2019], object detection [Ghiasi et al., 2020][Lehner et al., 2019], video editing [Lu et al., 2020], question answering [Yamada et al., 2020][Yamada et al., 2020], natural language processing [Brown et al., 2020], and many more. Thanks to the rise of generative models, results in the field of super-resolution [Sun and Chen, 2020][Chadha et al., 2020], image synthesis [Wang et al., 2020][Esser et al., 2020][Ramesh et al., 2021], text generation [Brown et al., 2020][Malmi et al., 2019], and countless more amaze scientists with accuracy and attention to details.

ANN also found their way into fields which traditional machine learning methods have not consider – fluid simulation [Um et al., 2018][Kim et al., 2019], cloth simulation [Lee et al., 2019][Sánchez-Banderas et al., 2020], or even whole physics [Holden et al., 2019][Sanchez-Gonzalez et al., 2020] and movement of virtual entities [Bergamin et al., 2019][Zhang et al., 2020]. Because of that, ANN are used in healthcare [Fakoor et al., 2013][McKinney et al., 2020], Hollywood [Rayo, 2019], robotics [Pierson and Gashler, 2017][Lee et al., 2020], and learning computer games [OpenAI et al., 2019][Vinyals et al., 2019].

This long and incomplete list of successes would not be possible without the computation power of computers and data centers nowadays. In fact, the achievements of DANN in 2011 and 2012 were driven mainly by the effective implementation of Convolutional Neural Networks (CNN) in kernels and running them inside the Graphical Processing Unit (GPU) [Cireşan et al., 2012]. Since then, the GPU and more recently the Tensor Processing Unit (TPU) have overcome traditional processors in terms of usage in ANN and allowed these advancements in the field we can see now.

Another interesting field of Artificial intelligence is evolutionary computation. Similarly to ANN, it performs parameters search in order to best solve the task in hand. Evolutionary algorithms and their variations have been successfully applied in variety of fields – for example electronic circuits design [Hornby et al., 2006] [Barari et al., 2014], real-parameter optimization [Wierstra et al., 2008], combinatorial problems [Anderson and Ferris, 1994][Albayrak and Allahverdi, 2011],

. Citace možná staré?

robotics [Bongard, 2013][Rieffel et al., 2009][Nygaard et al., 2018], design of neural networks [Stanley and Miikkulainen, 2002][Floreano et al., 2008], and many more.

There has been attempts to implement and run evolutionary algorithms on GPU [Cheng and Gen, 2019a][Cheng and Gen, 2019b] and use the computation power available. However, most of these methods focus on Compute Unified Device Architecture (CUDA) or similar technology. CUDA and alike technologies have drawbacks – they use low-end languages like C or C++, require deep knowledge of the underlying hardware and algorithms are harder to modify.

Goal of this work is therefore to analyze existing implementations of evolutionary algorithms and propose their implementations on GPU. The implementation needs to be easy to understand, extendable, and easily to modify. In order to meet given criteria, implementation should reuse existing frameworks available for neural networks if possible. Zmínit cíle explicitně, indikuje přímo závěr

The rest of this thesis is organized as follows. In the next Section, the evolutionary algorithms and their variants are introduced. Section 3 describes CUDA programming in order to design efficient implementation. Proposed implementation and design decisions are given in Section 4. In Section 5, set of problems is presented for the purpose of test, comparison and benchmark of the implementation. Section 6 evaluates the proposed implementation and compare it to standard methods. Finally, conclusion and main ideas are given in section 7.

https://www.shark-ml.org/sphinx_pages/build/html/rest_sources/tutorials/tutorials.html
http://paradiseo.gforge.inria.fr/

TSP lze řešit ant colony

BBOB multikriteriální optimalizace

Vícekriteriální optimalizace lepší než TSP

Vícekriteriální MOEA/D, NSGA-II
https://www.researchgate.net/publication/289937621_Parallel_MOEAD-ACO_on_GPU

2. Conclusion

Some conclusion.

Bibliography

[Albayrak and Allahverdi, 2011] Albayrak, M. and Allahverdi, N. (2011). Development a new mutation operator to solve the traveling salesman problem by aid of genetic algorithms. *Expert Systems with Applications*, 38(3):1313–1320.

[Anderson and Ferris, 1994] Anderson, E. and Ferris, M. (1994). Genetic algorithms for combinatorial optimization: The assemble line balancing problem. *INFORMS Journal on Computing*, 6:161–173.

[Barari et al., 2014] Barari, M., Karimi, H. R., and Razaghian, F. (2014). Analog circuit design optimization based on evolutionary algorithms. *Mathematical Problems in Engineering*, 2014:1–12.

[Bergamin et al., 2019] Bergamin, K., Clavet, S., Holden, D., and Forbes, J. R. (2019). DReCon: Data-driven responsive control of physics-based characters. *ACM Trans. Graph.*, 38(6).

[Bongard, 2013] Bongard, J. C. (2013). Evolutionary robotics. *Commun. ACM*, 56(8):74–83.

[Brown et al., 2020] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners.

[Chadha et al., 2020] Chadha, A., Britto, J., and Roja, M. M. (2020). iSeeBetter: Spatio-temporal video super-resolution using recurrent generative back-projection networks. *Computational Visual Media*, 6(3):307–317.

[Cheng and Gen, 2019a] Cheng, J. R. and Gen, M. (2019a). Accelerating genetic algorithms with GPU computing: A selective overview. *Computers & Industrial Engineering*, 128:514–525.

[Cheng and Gen, 2019b] Cheng, J. R. and Gen, M. (2019b). Accelerating genetic algorithms with GPU computing: A selective overview. *Computers & Industrial Engineering*, 128:514–525.

[Cireşan et al., 2012] Cireşan, D., Meier, U., Masci, J., and Schmidhuber, J. (2012). Multi-column deep neural network for traffic sign classification. *Neural Networks*, 32:333–338. Selected Papers from IJCNN 2011.

[Esser et al., 2020] Esser, P., Rombach, R., and Ommer, B. (2020). Taming transformers for high-resolution image synthesis.

[Fakoor et al., 2013] Fakoor, R., Ladhak, F., Nazi, A., and Huber, M. (2013). Using deep learning to enhance cancer diagnosis and classification. In *Proceedings of the international conference on machine learning*, volume 28. ACM, New York, USA.

[Floreano et al., 2008] Floreano, D., Dürr, P., and Mattiussi, C. (2008). Neuroevolution: from architectures to learning. *Evolutionary Intelligence*, 1:47–62.

[Ghiasi et al., 2020] Ghiasi, G., Cui, Y., Srinivas, A., Qian, R., Lin, T.-Y., Cubuk, E. D., Le, Q. V., and Zoph, B. (2020). Simple copy-paste is a strong data augmentation method for instance segmentation.

[Heck et al., 2000] Heck, L. P., Konig, Y., Sönmez, M., and Weintraub, M. (2000). Robustness to telephone handset distortion in speaker recognition by discriminative feature design. *Speech Communication*, 31(2):181–192.

[Holden et al., 2019] Holden, D., Duong, B. C., Datta, S., and Nowrouzezahrai, D. (2019). Subspace neural physics: Fast data-driven interactive simulation. In *Proceedings of the 18th Annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '19, New York, NY, USA. Association for Computing Machinery.

[Hornby et al., 2006] Hornby, G., Globus, A., Linden, D., and Lohn, J. (2006). Automated antenna design with evolutionary algorithms. *Collection of Technical Papers - Space 2006 Conference*, 1.

[Kim et al., 2019] Kim, B., Azevedo, V. C., Thuerey, N., Kim, T., Gross, M., and Solenthaler, B. (2019). Deep fluids: A generative network for parameterized fluid simulations. *Computer Graphics Forum*, 38(2):59–70.

[Lecun et al., 1998] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278 – 2324.

[Lee et al., 2020] Lee, J., Hwangbo, J., Wellhausen, L., Koltun, V., and Hutter, M. (2020). Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5(47):eabc5986.

[Lee et al., 2019] Lee, T. M., Oh, Y. J., and Lee, I.-K. (2019). Efficient cloth simulation using miniature cloth and upscaling deep neural networks.

[Lehner et al., 2019] Lehner, J., Mitterecker, A., Adler, T., Hofmarcher, M., Nessler, B., and Hochreiter, S. (2019). Patch refinement – localized 3D object detection.

[Lu et al., 2020] Lu, E., Cole, F., Dekel, T., Xie, W., Zisserman, A., Salesin, D., Freeman, W. T., and Rubinstein, M. (2020). Layered neural rendering for retiming people in video.

[Malmi et al., 2019] Malmi, E., Krause, S., Rothe, S., Mirylenka, D., and Severyn, A. (2019). Encode, tag, realize: High-precision text editing.

[McKinney et al., 2020] McKinney, S., Sieniek, M., Godbole, V., Godwin, J., Antropova, N., Ashrafian, H., Back, T., Chesus, M., Corrado, G., Darzi, A., Etemadi, M., Garcia-Vicente, F., Gilbert, F., Halling-Brown, M., Hassabis, D., Jansen, S., Karthikesalingam, A., Kelly, C., King, D., and Shetty, S. (2020). International evaluation of an AI system for breast cancer screening. *Nature*, 577:89–94.

[Nygaard et al., 2018] Nygaard, T. F., Martin, C. P., Samuelsen, E., Torresen, J., and Glette, K. (2018). Real-world evolution adapts robot morphology and control to hardware limitations.

[OpenAI et al., 2019] OpenAI, :, Berner, C., Brockman, G., Chan, B., Cheung, V., Debiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., Józefowicz, R., Gray, S., Olsson, C., Pachocki, J., Petrov, M., d. O. Pinto, H. P., Raiman, J., Salimans, T., Schlatter, J., Schneider, J., Sidor, S., Sutskever, I., Tang, J., Wolski, F., and Zhang, S. (2019). Dota 2 with large scale deep reinforcement learning.

[Pham et al., 2021] Pham, H., Dai, Z., Xie, Q., Luong, M.-T., and Le, Q. V. (2021). Meta pseudo labels.

[Pierson and Gashler, 2017] Pierson, H. A. and Gashler, M. S. (2017). Deep learning in robotics: A review of recent research.

[Ramesh et al., 2021] Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (2021). Zero-shot text-to-image generation.

[Rayo, 2019] Rayo, E. A. (2019). Artificial intelligence at Disney, Viacom, and other entertainment giants. `https://emerj.com/ai-sector-overviews/ai-at-disney-viacom-and-other-entertainment-giants/`. Accessed 2020-03-10.

[Rieffel et al., 2009] Rieffel, J., Saunders, F., Nadimpalli, S., Zhou, H., Hassoun, S., Rife, J., and Trimmer, B. (2009). Evolving soft robotic locomotion in PhysX. In *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*, GECCO '09, page 2499–2504, New York, NY, USA. Association for Computing Machinery.

[Sanchez-Gonzalez et al., 2020] Sanchez-Gonzalez, A., Godwin, J., Pfaff, T., Ying, R., Leskovec, J., and Battaglia, P. W. (2020). Learning to simulate complex physics with graph networks.

[Sánchez-Banderas et al., 2020] Sánchez-Banderas, R. M., Rodríguez, A., Barreiro, H., and Otaduy, M. A. (2020). Robust Eulerian-on-Lagrangian Rods. *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH)*, 39(4).

[Stanley and Miikkulainen, 2002] Stanley, K. O. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evol. Comput.*, 10(2):99–127.

[Sun and Chen, 2020] Sun, W. and Chen, Z. (2020). Learned image downscaling for upscaling using content adaptive resampler. *IEEE Transactions on Image Processing*, 29:4027–4040.

[Um et al., 2018] Um, K., Hu, X., and Thuerey, N. (2018). Liquid splash modeling with neural networks.

[Vinyals et al., 2019] Vinyals, O., Babuschkin, I., Czarnecki, W., Mathieu, M., Dudzik, A., Chung, J., Choi, D., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J., Jaderberg, M., and Silver, D. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575.

[Wang et al., 2020] Wang, L., Chen, W., Yang, W., Bi, F., and Yu, F. (2020). A state-of-the-art review on image synthesis with generative adversarial networks. *IEEE Access*, PP:1–1.

[Wierstra et al., 2008] Wierstra, D., Schaul, T., Peters, J., and Schmidhuber, J. (2008). Natural evolution strategies. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 3381–3387.

[Yamada et al., 2020] Yamada, I., Asai, A., Shindo, H., Takeda, H., and Matsumoto, Y. (2020). LUKE: Deep contextualized entity representations with entity-aware self-attention.

[Zawadzka-Gosk et al., 2019] Zawadzka-Gosk, E., Wołk, K., and Czarnowski, W. (2019). *Deep Learning in State-of-the-Art Image Classification Exceeding 99% Accuracy*, pages 946–957. Springer International Publishing.

[Zhang et al., 2020] Zhang, H., Sciutto, C., Agrawala, M., and Fatahalian, K. (2020). Vid2Player: Controllable video sprites that behave and appear like professional tennis players.

List of Figures

List of Tables

Acronyms

AI Artificial intelligence.

ANN Artificial Neural Networks.

CNN Convolutional Neural Networks.

CUDA Compute Unified Device Architecture.

DANN Deep Artificial Neural Networks.

GPU Graphical Processing Unit.

TPU Tensor Processing Unit.

A. Attachments

A.1 First Attachment

//TODO make sure attachments check