SpringerLink

Programs Features Clustering to Find Optimization Sequence Using Genetic Algorithm

International Conference on Information, Communication and Computing Technology

ICICCT 2019: Intelligent Computing Paradigm and Cutting-edge Technologies pp 40-50 | Cite as

- Manal H. Almohammed (1) Email author (Manalalmohamed.h@gmail.com)
- Esraa H. Alwan (1)
- Ahmed B. M. Fanfakh (1)

1. Department of Computer Science, College of Science for Women, University of Babylon, , Hillah, Iraq

Conference paper
First Online: 18 January 2020

- 2 Citations
- 170 Downloads

Part of the Learning and Analytics in Intelligent Systems book series (LAIS, volume 9)

Abstract

Finding the best optimization sequence order that can improve the performance for even a simple program is not an easy task. However, the modern compilers provide dozens of optimizations, making it not a practical solution to try all the optimization sequences manually to find the optimal one. In this paper, genetic algorithm is proposed to select the best optimization sequence for a cluster of similar programs. However, wide set of programs are elected to cover as much as possible all the features. The set of the programs are classified into three clusters depending on them features. Thus, the genetic algorithm in this work is learning method. This means any new program, unseen program, can take the optimization sequence of the cluster that has similar features to it. Moreover, two scenarios are proposed using genetic algorithm to find the best optimization sequence for each cluster. In the first scenario, programs are classified into three clusters according to program dynamic features. The genetic algorithm with Tournament selection method is applied on each cluster independently to obtain a good optimization sequence for a cluster. Moreover, the proposed method improved the execution time on average by 77% compared with the O2. The second scenario was exactly similar to the first one. While, different selection methods are used for each cluster. The improved average execution time for this scenario was 78% compared with the O2. LLVM framework is used to validate and execute the proposed method. In addition, Bolybench, Standerford, Shootout benchmarks are used to verify the effectiveness of the proposed method.

Keywords

Genetic algorithm LLVM Phase-ordering Optimizations

References

1. Shouri, A.H., Bignoli, A., Palermo, G., Silvano, C., Kulkarni, S., Cavazos, J.: Mitigating the compiler phase-ordering problem using optimization sub-sequences and machine learning. ACM Trans. Archit. Code Optim. **14**(3), 1–28 (2017)
   Google Scholar (http://scholar.google.com/scholar_lookup?title=Mitigating%20the%20compiler%20phase-ordering%20problem%20using%20optimization%20sub-sequences%20and%20machine%20learning&author=AH.%20Shouri&author=A.%20Bignoli&author=G.%20Palermo&author=C.%20Silvano&author=S.%20Kulkarni&author=J.%20Cavazos&journal=ACM%20Trans.%20Archit.%20Code%20Optim.&volume=14&issue=3&pages=1-28&publication_year=2017)
2. De Lima, E.D., de Souza Xavier, T.C., da Silva, A.F., Ruiz, L.B.: Compiling for Performance and Power Efficiency (2013)
   Google Scholar (https://scholar.google.com/scholar?q=De%20Lima%2C%20E.D.%2C%20de%20Souza%20Xavier%2C%20T.C.%2C%20da%20Silva%2C%20A.F.%2C%20Ruiz%2C%20L.B.%3A%20Compiling%20for%20Performance%20and%20Power%20Efficiency%20%282013%29)
3. Cavazos, J., Fursin, G., Agakov, F., Bonilla, E., O'Boyle, M.F., Temam, O.: Rapidly Selecting Good Compiler Optimizations using Performance Counters (2007)
   Google Scholar (https://scholar.google.com/scholar?q=Cavazos%2C%20J.%2C%20Fursin%2C%20G.%2C%20Agakov%2C%20F.%2C%20Bonilla%2C%20E.%2C%20O%E2%80%99Boyle%2C%20M.F.%2C%20Temam%2C%20O.%3A%20Rapidly%20Selecting%20Good%20Compiler%20Optimizations%20using%20Performance%20Counters%20%282007%29)
4. Alkaaby, Z.S., Alwan, E.H., Fanfakh, A.B.M.: Finding a good global sequence using multi-level genetic algorithm. J. Eng. Appl. Sci. **13**(22), 9777–9783 (2018)
   Google Scholar (http://scholar.google.com/scholar_lookup?title=Finding%20a%20good%20global%20sequence%20using%20multi-level%20genetic%20algorithm&author=ZS.%20Alkaaby&author=EH.%20Alwan&author=ABM.%20Fanfakh&journal=J.%20Eng.%20Appl.%20Sci.&volume=13&issue=22&pages=9777-9783&publication_year=2018)
5. Cavazos, J.: Mitigating the compiler optimization phase-ordering problem using machine learning. ACM SIGPLAN Not. **47**(10), 147–162 (2012)
   CrossRef (https://doi.org/10.1145/2398857.2384628)

Google Scholar (http://scholar.google.com/scholar_lookup?title=Mitigating%20the%20compiler%20optimization%20phase-ordering%20problem%20using%20machine%20learning&author=J.%20Cavazos&journal=ACM%20SIGPLAN%20Not.&volume=47&issue=10&pages=147-162&publication_year=2012)

6. Purini, S., Jain, L.: Finding good optimization sequences covering program space. ACM Trans. Archit. Code Opt. **9**(4), 56 (2013)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Finding%20good%20optimization%20sequences%20covering%20program%20space&author=S.%20Purini&author=L.%20Jain&journal=ACM%20Trans.%20Archit.%20Code%20Opt.&volume=9&issue=4&pages=56&publication_year=2013)

7. Majumder, A., Ekbal, A., Naskar, S. K.: Feature Selection and Class-Weight Tuning Using Genetic Algorithm for Biomolecular Major Steps for Event Extraction (2017)
Google Scholar (https://scholar.google.com/scholar?q=Majumder%2C%20A.%2C%20Ekbal%2C%20A.%2C%20Naskar%2C%20S.%20K.%3A%20Feature%20Selection%20and%20Class-Weight%20Tuning%20Using%20Genetic%20Algorithm%20for%20Biomolecular%20Major%20Steps%20for%20Event%20Extraction%20%282017%29)

8. Trivedi, A., Srinivasan, D., Biswas, S., Reindl, T.: Hybridizing genetic algorithm with differential evolution for solving the unit commitment scheduling problem, Swarm Evol. Comput. 1–15 (2015)
Google Scholar (https://scholar.google.com/scholar?q=Trivedi%2C%20A.%2C%20Srinivasan%2C%20D.%2C%20Biswas%2C%20S.%2C%20Reindl%2C%20T.%3A%20Hybridizing%20genetic%20algorithm%20with%20differential%20evolution%20for%20solving%20the%20unit%20commitment%20scheduling%20problem%2C%20Swarm%20Evol.%20Comput.%201%E2%80%9315%20%282015%29%0A)

9. Davidson, J.W., Tyson, G.S., Whalley, D.B., Kulkarni, P.A.: Evaluating Heuristic Optimization Phase Order Search Algorithms (2007)
Google Scholar (https://scholar.google.com/scholar?q=Davidson%2C%20J.W.%2C%20Tyson%2C%20G.S.%2C%20Whalley%2C%20D.B.%2C%20Kulkarni%2C%20P.A.%3A%20Evaluating%20Heuristic%20Optimization%20Phase%20Order%20Search%20Algorithms%20%282007%29)

10. Muslim, A.B., Ali, A.K.M.: The combination of genetic programming and genetic algorithm for neural networks design and training. J. Univ. Babylon **18**(1–2), 350–359 (2011)
Google Scholar (http://scholar.google.com/scholar_lookup?title=The%20combination%20of%20genetic%20programming%20and%20genetic%20algorithm%20for%20neural%20networks%20design%20and%20training&author=AB.%20Muslim&author=AKM.%20Ali&journal=J.%20Univ.%20Babylon&volume=18&issue=1%E2%80%932&pages=350-359&publication_year=2011)

11. Lattner, C., Adve, V.: LLVM: A compilation framework for lifelong program analysis & transformation. In: Proceedings of the International Symposium on Code Generation and Optimization IEEE Computer Society (2004)
Google Scholar (https://scholar.google.com/scholar?q=Lattner%2C%20C.%2C%20Adve%2C%20V.%3A%20LLVM%3A%20A%20com

pilation%20framework%20for%20lifelong%20program%20analysis%20%26%20transformation.%20In%3A%20Proceedings%20of%20the%20International%20Symposium%20on%20Code%20Generation%20and%20Optimization%20IEEE%20Computer%20Society%20%282004%29)

12. Pan, Z., Eigenmann, R.: Fast and effective orchestration of compiler optimizations for automatic performance tuning. In: Proceedings of the International Symposium on Code Generation and Optimization IEEE Computer Society, 319–332 (2006)
Google Scholar (https://scholar.google.com/scholar?q=Pan%2C%20Z.%2C%20Eigenmann%2C%20R.%3A%20Fast%20and%20effective%20orchestration%20of%20compiler%20optimizations%20for%20automatic%20performance%20tuning.%20In%3A%20Proceedings%20of%20the%20International%20Symposium%20on%20Code%20Generation%20and%20Optimization%20IEEE%20Computer%20Society%2C%20319%E2%80%93332%20%282006%29)

Copyright information

About this paper

Cite this paper as:

Almohammed M.H., Alwan E.H., Fanfakh A.B.M. (2020) Programs Features Clustering to Find Optimization Sequence Using Genetic Algorithm. In: Jain L., Peng SL., Alhadidi B., Pal S. (eds) Intelligent Computing Paradigm and Cutting-edge Technologies. ICICCT 2019. Learning and Analytics in Intelligent Systems, vol 9. Springer, Cham. https://doi.org/10.1007/978-3-030-38501-9_4

- First Online 18 January 2020
- DOI https://doi.org/10.1007/978-3-030-38501-9_4
- Publisher Name Springer, Cham
- Print ISBN 978-3-030-38500-2
- Online ISBN 978-3-030-38501-9
- eBook Packages Intelligent Technologies and Robotics Intelligent Technologies and Robotics (R0)

Personalised recommendations