Evolving Compiled Code

Eric C. Collom
University of Minnesota, Morris
coll0474@morris.umn.edu

ABSTRACT

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Delphi theory

Keywords

ACM proceedings, LaTeX, text tagging

1. INTRODUCTION

The limitation of only being able to evolve small parts of programs is an issue in evolutionary computation (EC) am I BS'ing or is this legit?. Much of these constraints are due to there is no graceful way to evolve programs, as a whole, directly at the source code. This is because source code exists to make it easier for humans to read and write code. However, it is very hard to create a genetic algorithm (GA) that would evolve source code. Even if one was created it would basically be the size of a compiler Need to review the compiler argument in finch. feel like I'm BS'ing. A solution to this problem is to to compile the program into a lower level language and evolve it there. Two lower level languages at which this has been done is Java bytecode and x86 assembly [1] [2] which I will discuss thoroughly in this paper add more.

2. THE *BODY* OF THE PAPER

3. CONCLUSIONS

4. ACKNOWLEDGMENTS

UMM CSci Senior Seminar Conference, April 2014 Morris, MN.

5. REFERENCES

[1] M. Orlov and M. Sipper. Flight of the finch through the java wilderness. *Evolutionary Computation, IEEE Transactions on*, 15(2):166–182, April 2011.
This will be one of the backbones of my paper. Talks about a program called FINCH that the authors developed. Finch is used to evolve programs at the Java bytecode level. This paper Has lots of information and they have about six examples where they evolved programs using FINCH .

[2] E. Schulte, S. Forrest, and W. Weimer. Automated program repair through the evolution of assembly code. In *Proceedings of the IEEE/ACM International Conference on Automated Software Engineering*, ASE '10, pages 313–316, New York, NY, USA, 2010. ACM.
This will be a main source for my paper. It talks about evolving programs at assembly language level and its benefits. These authors focus more on doing this to use EC to debug, refactor, and repair programs. It also breifly mentions FINCH .