Pedocomputing: A Generalized Framework for Polynomial-Time Solution of NP Problems

Gary Yngve

May 17, 2002

Abstract

We introduce pedocomputing, a novel alternative computing paradigm. We show how pedocomputing can solve NP problems in polynomial time. We also show other uses of pedocomputing for efficient computation, including approximation algorithms, artificial intelligence, and computer graphics.

1 Introduction

One of the most active research areas in computer science is alternative computing paradigms. However, much of these paradigms are fantasies at best: quantum computing and DNA computing are far from being able to solve problems of actual interest to industry, and as for necrocomputing[Underwood], well, that is just fantasy. We introduce pedocomputing, a paradigm with immediate application to a huge range of problems. Pedocomputing, through parallelism with an exponential number of children, can solve any problem in NP in polynomial time. The rest of the paper proceeds as follows. First we discuss previous work. Then we present the principles behind pedocomputing and its efficiency. Then we show how pedocomputing can be used to solve hard problems, find approximate solutions to hard problems, and produce outstanding results for classic AI and graphics problems.

2 Previous Work

The idea of using children for work or other uses has been known for many years. Jonathan Swift proposed using children for food in the 1700s. Although many scholars were skeptical and did not believe his proposal was serious, and Swift insisted on being modest and not capitalizing on his brilliant idea, most people at the time were quite alarmed, perhaps because his ideas were so futuristic. Gifford, et al., have recently demonstrated the economic advantages of using children. Major corporations such as Nike take advantage of distributed systems, spreading their labor among children.
children all around the world. Everquest has a brilliant marketing scheme of charging people to do work. However, the work that the people perform is mostly useless. Perhaps the oldest known use of children has been by the Catholic Church. As with many nonacademic institutions, the extent and timeframe of their research is largely unknown, but recently their work has come to the limelight.

3 Pedocomputing: The Theory

The theory behind pedocomputing is very simple. Pedocomputing is just a nondeterministic Turing machine. It takes polynomial time to create an exponential number of kids. An exponential number of kids is an exponential number of pedoflops. It likewise takes polynomial time to distribute state information to the kids. In polynomial time, the kids can be taught to verify their solutions and report successful ones. Hence any NP problem can be solved by an exponential number of kids in polynomial time.

4 Graph k-Coloring: Randomization and Approximation

The decision problem of whether or not a graph is k-colorable is an NP-complete problem [Garey and Johnson]. Although the previous section clearly shows that any NP problem can be solved with pedocomputing, we show that even a small number of kids can efficiently solve the k-coloring problem. A graph is presented to children as a picture, and the kids color them (figure 1). Each child imparts his or her own creativity and randomness into the picture, so with high probability, if a k-coloring exists, it will be found if enough children color. An advantage of our method over previous work is that the children seem to enjoy coloring much more than they enjoy sewing sneakers. We can get even faster results by using younger kids. They color pages much faster, but with greater error. For example, in the second image of figure 1, the k-coloring is accurate to $\varepsilon$, where $\varepsilon$ is 3 inches. Unfortunately, we have found that pedocomputing is not as effective as we liked when used with older kids (third image). Currently our k-coloring solution is limited to $k=120$, the most different colors of crayons we have found thus far. We are working with industry affiliates such as Crayola to increase the number of colors.

5 AI and Graphics

Pedocomputing also has applications to artificial intelligence and computer graphics. For years, researchers have been trying to reproduce the mental capacity of a five-year-old in a computer and have failed miserably. However, pedocomputing with five million five-year-olds has roughly the mental capacity of five million five-year-olds.

Pedocomputing also has applications to nonphotorealistic rendering in graphics. Figure 2 shows the state-of-the-art NPR filters
Figure 1: Correct k-coloring solution by a five-year-old.

Figure 2: Nonphotorealistic renditions of a photograph by traditional computing and pedocomputing.
in Adobe Photoshop compared with the NPR filter produced by pedocomputing. There is simply no comparison: pedocomputing produces far more artistic effects.

6 Conclusions

Pedocomputing is a novel alternative computing paradigm that allows for polynomial-time solution of NP problems by modeling a nondeterministic Turing machine with children. It also is able to solve certain classes of problems efficiently or approximately without an exponential number of children. However, there are limitations to pedocomputing. Classic parallel algorithms such as Fast Fourier Transform do not work well with pedocomputing. We performed a study of a Cray versus a pedocomputing cluster in India. The Cray outperformed the pedocomputing cluster on the Fast Fourier Transform except for the performance of a three-year-old purportedly a descendant of Ramanujan.

7 Acknowledgements and Apologies

We would like to acknowledge Eugene Hsu for finding the appropriate childish mentalities and assisting with his Photoshop abilities. We would also like to thank various lounge denizens (e.g. Albert_{0}) for their helpful comments on this paper. We are not thanking other lounge denizens (e.g. Albert') for their obscene comments on this paper. We would like to apologize to the Catholic Church for any possible offense. But remember that even God has a sense of humor — just look at the platypus. We would like to apologize to all platypuses for any possible offense [Smith]. We also would like to apologize to Hannah Tang, who insists her confusion from reading a theory paper had nothing to do with the actual content but was because we reordered her pages.