Journal of Uncertain Systems
Vol.7, No.3, pp.238-240, 2013
Online at: www.jus.org.uk

# A New Analog Optical Processing Scheme for Solving NP-Hard Problems

Michael Zakharevich[1], Vladik Kreinovich[2,*]

[1] *Aligh Technology Inc.*

[2] *Department of Computer Science, University of Texas at El Paso, El Paso, Texas 79968, USA*

Received 2 April 2012; Revised 25 August 2012

**Abstract**

Many real-life problems are, in general, NP-hard, i.e., informally speaking, are difficult to solve. To be more precise, a problem $\mathcal{P}_0$ is NP-hard means that every problem from the class NP can be reduced to this problem $\mathcal{P}_0$. Thus, if we have an efficient algorithm for solving one NP-hard problem, we can use this reduction to get a more efficient way of solving all the problems from the class NP. To speed up computations, it is reasonable to base them on the fastest possible physical process – i.e., on light. It is known that analog optical processing indeed speeds up computation of several NP-hard problems. Each of the corresponding speed-up schemes has its success cases and limitations. The more schemes we know, the higher the possibility that for a given problem, one of these schemes will prove to be effective. Motivated by this argument, we propose a new analog optical processing scheme for solving NP-hard problems.

**Keywords:** NP-hard problems, optical computing, analog computing

## 1 Introduction

**Solving NP-hard problems is important.** In practice, we often to find a solution that satisfies a given set of constraints. Once we have a candidate for the solution, we can feasibly check whether this candidate indeed satisfies all the constraints. In theoretical computer science, "feasibly" is usually interpreted as computable in polynomial time, i.e., in time bounded by a polynomial of the length of the input. The class of all problems for which we can check, in polynomial time, whether a given candidate is a solution, is known as the class NP; see, e.g., [3].

Examples of such problem includes coloring a given graph, finding the values of propositional variables $x_1, \ldots, x_n$ that satisfy a given propositional formula, i.e., formula of the type

$$(x_1 \vee \neg x_2 \vee x_3) \,\&\, (x_4 \vee \neg x_2 \vee \neg x_5) \,\&\, \cdots, \text{ etc.}$$

Each problem from the class NP can be algorithmically solved by trying all possible candidates. For example, we can find a graph coloring by trying all possible assignments of colors to different vertices of a graph, and we can find a satisfying propositional vector by trying all $2^n$ possible combinations of true and false values $x_1, \ldots, x_n$. Such exhaustive search algorithms require computation time like $2^n$, time that grows exponentially with $n$. For medium-size inputs, e.g., for $n \approx 300$, the resulting time is larger than the lifetime of the Universe. So, these exhaustive search algorithms are not practically feasible.

It is not known whether problems from the class NP can be solved feasibly (i.e., in polynomial time): this is a famous open problem P$\overset{?}{=}$NP. It is known, however, there are problems which are NP-hard in the sense that every problem from the class NP can be reduced to this problem. Reduction means that if we can find a way to efficiently solve one NP-hard problem, then, by reducing other problems from the class NP to this problem, we can thus efficiently solve all the problems from the class NP.

So, it is very important to be able to efficiently solve even one NP-hard problem – no matter how exotic and unusual this problem may look. (By the way, both above example of NP problems – coloring a graph and finding the values of propositional variables – are NP-hard.)

[*] Corresponding author. Email: vladik@utep.edu (V. Kreinovich).

*Comment.* Several possible techniques for solving NP-hard problems are described in [1]; see also [4].

**Analog optical processing as a natural way to speed up solution of NP-hard problems.** A natural way to speed up computations is based on the fact that computations do involve actually moving data from one location to another. To speed up computations, it is therefore reasonable to move data as fast as possible. According to modern physics, the fastest possible process is light. So, to speed up computations, it is reasonable to use light, i.e., to perform actual *optical* processing.

Another natural way to speed up data processing is to parallelize computations. The more processors are available to work in parallel, the faster they can solve the original problem. Thus, instead of using a single ray of light, we can use numerous photons going in parallel, i.e., in effect, use a transmitted *image*.

**Analog optical processing can indeed solve NP-hard problems.** It is known that analog optical processing can indeed solve NP-hard problems; see, e.g., [5, 6] and references therein. This does not mean, of course, that we can actually solve NP-hard problems in polynomial time: polynomial time corresponds to the ideal case when all analog operations are performed perfectly well, with no errors or noise, and in reality, there is always noise. However, while we may not achieve polynomial time, we often do achieve a significant speed-up due to natural parallelism of optical data processing [5, 6].

**New schemes are always welcome.** The experience of the existing schemes of analog optical processing shows that each successful schemes has a niche in which it is practically helpful, and it has its limitations – i.e., problems for which it does not perform that well. From this viewpoint, the more schemes we propose, the larger the chances that one of these schemes will help to solve a future challenging problem.

**What we do in this paper.** Motivated by the above argument, in this paper, we propose a new analog optical scheme for solving NP-hard problems.

## 2 Proposal

**Selecting an NP-hard problem.** Out of many NP-hard problems described, e.g., in [2], we selected the one whose implementation seems to be the closest to optical computing. Namely, we selected the following *cosine product integration problem*: given a sequence of integers $a_1, \ldots, a_n$, check whether the integral

$$\int_0^{2\pi} \left( \prod_{i=1}^n \cos(a_i \cdot \theta) \right) d\theta$$

is equal to 0.

**How this problem can be implemented in analog optical processing.** To implement the above problem in terms of optical computing, we need to be able to do the following:

- implement $\cos(a_i \cdot \theta)$ for a given integer $a_i$,
- implement the product of two functions, and
- implement the integration.

In all these implementations, we will use ideas described in [5, 6].

Let us start by implementing cosines. We will represent each intermediate function $f(\theta)$ of $\theta$ as either a planar beam of width proportional to $2\pi$ or as a filter through which this beam passes. For the light beam, the value $f(\theta)$ represents the intensity of this beam at location $\theta$.

The intensity of the usual (non-polarized) light is a non-negative number. Since the value $\cos(a_i \cdot \theta)$ can be negative, we thus need (laser-generated) *coherent polarized* light, i.e., light that depends on time as $A \cdot \cos(\omega \cdot t + \varphi)$. In coherent optics, coherent light is usually described by a complex function $I \cdot \exp(\mathrm{i} \cdot \omega \cdot t)$, where $I \overset{\text{def}}{=} A \cdot \exp(\mathrm{i} \cdot \varphi)$. The value $I$ is positive when $\varphi = 0$ and negative when $\varphi = \pi$.

We start with light for which $I(\theta) = 1$ for all all spatial locations $\theta$.

From the physical viewpoint, light is an electromagnetic wave, i.e., an electromagnetic field traveling through space. The directions of the corresponding electric field $\vec{E}$ and magnetic field $\vec{H}$ are orthogonal to each other and to the direction of the light's propagation. Different polarizations mean different ways for the vectors $\vec{E}$ and $\vec{H}$ to change with time.

If we select the direction of the light propagation as the $z$-direction of the orthogonal coordinate system $(x, y, z)$, then the vectors $\vec{E}$ and $\vec{H}$ belong to the $(x, y)$-plane. Let us start with the *linearly polarized* light, for which $\vec{E}$ is always parallel to the $x$-axis, i.e., has the form $(E, 0, 0)$, and $\vec{H}$ is, therefore, always parallel to the $y$-axis.

There are physical processes that change the polarization, i.e., that rotate the vector $\vec{E}$ in the $(x, y)$-plane. It is possible to generate a spatial signal that will lead to a rotation angle proportional to $\theta$, namely, equal to $a_i \cdot \theta$. After the corresponding rotation, the light $(E(\theta), 0, 0)$ at a spatial location $\theta$ changes to $(E(\theta) \cdot \cos(a_i \cdot \theta), E(\theta) \cdot \sin(a_i \cdot \theta), 0)$. We can now place a polarization filter that only passes the the $x$-oriented light. Then, we get the light beam $(E(\theta) \cdot \cos(a_i \cdot \theta), 0, 0)$.

This scheme enables us to transform an $x$-oriented beam with intensity $E(\theta)$ into an $x$-oriented beam with intensity $E(\theta) \cdot \cos(a_i \cdot \theta)$. In other words, this scheme enables us to multiply the function representing the beam by $\cos(a_i \cdot \theta)$. So, if we start with a uniform beam for which $E(\theta) = 1$, and apply the above procedure $n$ times, with $i = 1, \ldots, n$, we get a light beam with intensity

$$I(\theta) = \prod_{i=1}^{n} \cos(a_i \cdot \theta).$$

Integration is also easy: all we have to do is collect all the light beam into a single point by using a focusing lens. The intensity of the resulting light is exactly

$$\int_0^{2\pi} \left( \prod_{i=1}^{n} \cos(a_i \cdot \theta) \right) d\theta.$$

So, by comparing the resulting intensity with 0, we can solve the original NP-hard integration problem.

## Acknowledgments

This work was supported in part by the National Science Foundation grants HRD-0734825 (Cyber-ShARE Center of Excellence) and DUE-0926721, by Grant 1 T36 GM078000-01 from the National Institutes of Health, and by a grant on F-transforms from the Office of Naval Research.

## References

[1] Aaronson, S., NP-complete problems and physical reality, *SIGACT New*, vol.36, pp.30–52, 2005.

[2] Garey, M.R., and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, New York, 1979.

[3] Papadimitriou, C., *Computational Complexity*, Addison Welsey, Reading, Massachusetts, 1994.

[4] Srikanth, R., The quantum measurement problem and physical reality: a computation theoretic perspective, *AIP Conference Proceedings*, vol.864, pp.178–193, 2006.

[5] Woods, D., and T.J. Naughton, An optical model of computation, *Theoretical Computer Science*, vol.334, pp.227–258, 2005.

[6] Woods, D., and T.J. Naughton, Optical computing, *Applied Mathematics and Computation*, vol.215, no.4, pp.1417–1430, 2009.