

How to read a scientific article

Liren Huang
Damianos Melidis

Faculty of Technology, Bielefeld University

huanglr at cebitec.uni-bielefeld.de
dmelidis at cebitec.uni-bielefeld.de

October 29, 2015

What is an scientific article

Breakthrough in the field

Different kinds of articles:

- Journal or conference proceedings
- Free or payed access
- Limited or unlimited number of pages
- Different review mechanisms
- Difficulty to publish depends on the quality of the article but also on the requirements of the publisher
- Different readers
- Theoretic results, practical results, discussion, review, letter
- Time between work and publication

Different kinds (content) of articles:

research article

- Exhaustive
- Objective

scientific review

- Comprehensive
- Subjective

Application Notes

- Implementations
- Simple and compact
- Practical

How to read an article

How to read a scientific article

First step: find an interesting article

- Recommendation by a friend or teacher
- Search from latest review paper
- Citation in another article
- Presented during a conference
- Web search to create or update your knowledge about a research field
- Articles which cite your work
- Journal club
- Articles with lots of citations
- New journal

How to read an article

Academic articles cannot be read effectively in the same way as a short story, a novel or a newspaper report.

An academic article is very dense, each sentence is important.
An article must be read several times, in different ways. And you have to find your ways to read it.

Gaming is a hard job, but someone has to do it!

GIOVANNI VIGLIETTA
Carleton University, Ottawa, Canada
viglietta@gmail.com

October 29, 2013

Abstract

We establish some general schemes relating the computational complexity of a video game to the presence of certain common elements or mechanics, such as destroyable paths, collectible items, doors opened by keys or activated by buttons or pressure plates, etc. Then we apply such “metatheorems” to several video games published between 1980 and 1998, including Pac-Man, Tron, Lode Runner, Boulder Dash, Delfektor, Mindbender, Pipe Mania, Skweek, Prince of Persia, Lemmings, Doom, Puzzle Bobble 3, and Starcraft. We obtain both new results, and improvements or alternative proofs of previously known results.

1 Introduction

This work was inspired mainly by the recent papers on the computational complexity of video games by Forišek [4] and Cormode [2], along with the excellent surveys on related topics by Kendall et al. [8] and Demaine et al. [3, 7], and may be regarded as their continuation on the same line of research.

Our purpose is to single out certain recurring features or mechanics in a video game that enable general reduction schemes from known hard problems to the games we are considering. To this end, in Section 2 we produce several *metatheorems* that will be applied in Section 3 to a wealth of famous commercial video games, in order to automatically establish their hardness with respect to certain computational complexity classes (with a couple of exceptions).

Because most recent commercial games incorporate Turing-equivalent scripting languages that easily allow the design of undecidable puzzles as part of the gameplay, we will focus primarily on older, “scriptless” games. Our selection includes games published between 1980 and 1998, presented in alphabetical order for better reference. Not every game will be rigorously explained in all its aspects

- **Material and methods:** description of algorithm
- **Results:** evaluation
- **Discussion/Conclusion:** conclusion and open work

Different way to read

Because we can read an article for different reasons, we don't read an article in the same way all the times, it also depends on the time we have for the reading.

What do you want to get out of it?

- **Read the whole article**
- Focus on the introduction (more references)
- Focus on the methods
- Focus on the conclusion
- Focus on the figures research field

Don't read everything the first time.

- 1 Take the time to really understand the **title**, then the **abstract** (and keywords).
- 2 Read the **introduction**, then the **conclusion**: understand the aim and the interest before the technique.
- 3 **Understand enough of the background.** Review what you already know about the topic.
- 4 **Read again** the introduction and conclusion.

Don't read everything the first time.

- 1 **Skim the method and result sections:** only the first sentence of each paragraph or only the figures and tables for example. Not the proof during the first run.
- 2 **Repeat reading** of the methods and results sections more and more deeply.
- 3 Read the **whole article**.

Take notes during the reading!

What is important? Not understood? Wrong? Nice? Inspiring?

Try to summarize the idea of each paragraph and express in your own words. Keep a clear landscape of the article structure

Questions before to start to read:

- Date?
- Journal article? In proceedings? Which journal/conference?
- Popularity (cited, see <http://citeseerx.ist.psu.edu/index>)
- Which field?
- Historical context?

Tips:

- Don't ignore the part that you don't understand
- There are mistakes (small or big...)
- Do not believe everything they say
- An article tells a story, you must be able to recount it
- Read cited papers
- Manage your time

How to read an article

- Is the topic interesting?
- Is it a new topic?
- What is the historical context? What is the impact today?
- What is the main point of this article?
- How is the quality of the writing? Is it clear? Are there mistakes?
- Is there sufficient background information?
- Are the evaluations and proofs sufficient(quantity and quality)?
- Did they forget state of the art? Explanation? Discussion?
- Do we have enough information to repeat their work?
- Which future work is suggested? Maybe it is already done...
- How to improve the presented work?

Thank you!

Questions?

Acknowledgements: Annelise Thevenin Fun with Algorithms Winter 2013-2014