

Great Ideas in Computing

University of Toronto CSC196
Fall 2021

Week 8: November 1-5 (2021)

Week 8 slides

Announcements:

- I have posted Professor Shah's slides for his discussion on fair division. Any thoughts on social choice?
- I have posted preliminary slides for week 8.
- I will soon be posting assignment 3. Please be sure that your assignments are properly uploaded onto Markus.

This weeks agenda

- New topic: Search engines

New topic: search engines

I think of search engines as a great idea in the sense of being a "killer application" and also leading to interesting computational issues that have energized the field of computing.

In doing so I am mostly talking about search engines as they are mainly used today in terms of searching web documents. That is, search engines that take queries (usually in the form of key words or phrases) and produce a *ranked list* of documents.

I am mostly going to talk about search engines independent of the importance (and necessity) of having large pools of fast machines, high speed communication and massive storage.

Search engines introduction continued

That is, I am mostly going to talk about search engines in terms of their functionality and the basic computational ideas that make them work (so) well. This is another example (like deep neural nets) of a great idea where greatness depended on new technology. Today's quality search engines simply could not exist say using the technology of the 1960s and 70s.

It is also an example where its greatness may also be an inhibitor for thinking about how to “significantly” move beyond the *current norm of key word based search*.

Of course, in some sense we have moved beyond just key word search in that one can now input an image and ask the search engine to find examples like that image.

Search engines introduction continued

In addition to ML being used for image recognition, ML is now playing a more algorithmic role in the quality of key word search.

But still from a functional point of view, while the quality of search has greatly improved, we are still basically doing what we did since say the late 1990s when say Google and Yahoo launched public search engines.

As we will see, the basic inspiration and ideas for current search engines has existed for quite some time.

A little search engine history

Search engines are part of the topic of "information retrieval" once the domain of library science. Computerized information retrieval has been an application idea since the start of modern computing.

On the web page there is a link to a prophetic July, 1945 Atlantic article "As We May Think" by Vannevar Bush where he envisions something quite close in many respects to the modern web and hyperlinked documents.

The article begins with the following: "Consider a future device . . . in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory."

That is, some kind of semi-automated information retrieval has been thought about for over 75 years.

Some quotes from the Vannevar Bush article

There are a lot of anachronisms (in terms of what the underlying technology will be, and gender roles) in this article but more important there are many insightful ideas about the future of accessing information. Here are some quotes from that article.

“Much needs to occur, however, between the collection of data and observations, the extraction of parallel material from the existing record, and the final insertion of new material into the general body of the common record. For mature thought there is no mechanical substitute. But creative thought and essentially repetitive thought are very different things. For the latter there are, and may be, powerful mechanical aids.”

Note: Bush is then clearly drawing his line here between human intelligence (and say creating new knowledge) vs retrieving existing knowledge.

More quotes from Bush's article in the Atlantic

“Our ineptitude in getting at the record is largely caused by the artificiality of systems of indexing. ... The human mind does not work that way. It operates by association. With one item in its grasp, it snaps instantly to the next that is suggested by the association of thoughts, in accordance with some intricate web of trails carried by the cells of the brain.”

“Man cannot hope fully to duplicate this mental process artificially, but he certainly ought to be able to learn from it. In minor ways he may even improve; e.g., so that for his records have relative permanency. The first idea, however, to be drawn from the analogy concerns selection. **Selection by association, rather than indexing**, may yet be mechanized.”

“Wholly new forms of encyclopedias will appear, ready made with a mesh of associative trails running through them, ready to be dropped into the **memex** and there amplified.” [Think now of hyperlinks.](#)

“Consider a future device for individual use, which is a sort of mechanized private file and library. It needs a name, and, to coin one at random, **memex** will do.”

The debate as to the nature of information retrieval

In the 1960's and 70's, there was a "debate" (albeit not widely discussed outside of those interested in information retrieval) between those who felt that information retrieval (IR) (i.e. finding documents to satisfy an "information need") was a subfield of AI (and more specifically natural language understanding) verses those who thought it could be best realized by more well established combinatorial, algebraic and statistical ideas. **Bush seems to have already settled his views well before this debate is taking place**

That is, one constituency felt that we needed to be able to "understand" what a document was saying (and what people were requesting) so as to find relevant documents.

The other constituency felt that the claims of many AI researchers were not at all feasible and that again a more statistical/algebraic/combinatorial approach (devoid of any real "intelligence") would produce better results.

The debate continued

I had a course (1967) in IR from Gerald Salton, who (according to Wikipedia) was "perhaps the leading computer scientist working in the field of information retrieval during his time". His group at Cornell developed the SMART Information Retrieval System".

I am not a great historian but I believe the vector space model (which we will discuss) was his idea. Salton was a proponent of the statistical/algebraic/combinatorial approach. I think that he always felt that AI was over-hyped.

So who won the debate?

The debate continued

I had a course (1967) in IR from Gerald Salton, who (according to Wikipedia) was "perhaps the leading computer scientist working in the field of information retrieval during his time". His group at Cornell developed the SMART Information Retrieval System".

I am not a great historian but I believe the vector space model (which we will discuss) was his idea. Salton was a proponent of the statistical/algebraic/combinatorial approach. I think that he always felt that AI was over-hyped.

So who won the debate?

As of today, it is clear that the approach of the constituency represented by Salton has turned out to be the basis for the way we currently do search in the internet.

The debate continued

I had a course (1967) in IR from Gerald Salton, who (according to Wikipedia) was "perhaps the leading computer scientist working in the field of information retrieval during his time". His group at Cornell developed the SMART Information Retrieval System".

I am not a great historian but I believe the vector space model (which we will discuss) was his idea. Salton was a proponent of the statistical/algebraic/combinatorial approach. I think that he always felt that AI was over-hyped.

So who won the debate?

As of today, it is clear that the approach of the constituency represented by Salton has turned out to be the basis for the way we currently do search in the internet. However, as I already indicated, search engines today do incorporate statistical ML into their retrieval algorithms.

Key word search

At a very general level, we can think of current search as the following process:

- 1 A user converts an “information need” into a query (i.e. a set of key words)
- 2 The search engine is then an algorithm for the mapping:
query \times {collection of documents} \rightarrow <ranked list of “relevant documents” >.
- 3 Upon receiving highly ranked documents, the user may choose to refine the query.
- 4 This process continues until the user is either satisfied or gives up.
How often do you have to refine your queries? In a given week, how many search do you do and how often do you abandon a search?

As we discuss the ideas behind key word search in search engines, it should be noted that there are many specific ideas and engine specific details that go into making a search engine successful (in terms of the quality, speed, and coverage) and these ideas and details are kept reasonably confidential.

Why?

Why the secrecy?

There are two main reasons for not disclosing specific search engine ideas and details:

- Not surprisingly, these ideas are trade secrets that give a company a competitive edge.
- Perhaps less obvious, knowing exactly how a company does its searches allows one to easily spam documents so as to raise their ranking (and hence lower the quality of the ranking).

So please be advised that what I am discussing is just the high level ideas and not the specifics say being utilized by Google, Yahoo or Microsoft.

It clearly took significant progress in technology (i.e., the speed and memory capabilities of large numbers of distributed machines) to make key word search as successful as it is today. Equally important, many significant algorithmic ideas plus extensive and ongoing experience with user requests has been necessary for search engine success.

However, collecting information from user interactions is, of course, an important privacy issue.

The challenge of real time information retrieval

In addition to algorithmic ideas used to improve search quality (i.e., precision, recall), commercial search engines are dealing with enormous collections of sites/documents and must return responses in what appears to be "real time".

Estimates of the size of the web vary. One site (WorldWideWebSize.com) provides daily reports on the size of the web: That site reported "The Indexed Web contains at least 5.42 billion pages (Sunday, 04 October, 2020)" but as of October 14, 2021, it reports "The Indexed Web contains at least 4.81 billion pages". **Did the size really decline?**

Precision and Recall

Let D be a collection of documents and S a set of documents being presented to a user. *Precision* in a set of documents (for an information need) is defined as the fraction of returned documents that are relevant. That is, if S has s relevant documents then precision is defined as $\frac{s}{|S|}$. In a ranked list we can informally say that precision means that the higher the rank of the document, the more likely it is to be relevant. (See the question in assignment 3.)

Recall in a set $S \subset D$ of returned documents is defined as the fraction of all relevant documents in the returned set; that is, $\frac{s}{d}$.

There is almost always a tradeoff between precision and recall. **Why?**

End of Monday, November 1 class

Today we will finish up the discussion of search engines. Then if there is time, we will start a discussion of complexity theory and in particular the (literally) million dollar question: Is $P = NP$ or $P \neq NP$.

Do we want diversity in the documents retrieved?

We may (or may not) want the highest ranked documents to reflect some desired diversity.

For example, what if I provide the query “What did Donald Trump accomplish as US president”? Do I want just what is reported as his positive accomplishments? Or do I want just the negative aspects of his presidency? Or do I want a diversity of opinions? Are all opinions equally meaningful? Do we want denials of the Holocaust to be presented in the name of diversity and “balance” as some in the Texas legislature demand? What is a “legitimate” opinion vs a conspiracy theory devoid of facts?

Similarly, if I ask whether the stock market has recently been rising? Do I want some overall assessment, or do I want reports on different sectors of the market?

More on diversity

Even for a more classical and now perhaps a more mundane example, when I ask for recent information about “jaguars”, do I mean the car, the animal or the NFL football team? I probably only want one of these. When I make my request clear, a search engine should avoid ambiguous meanings.

More on diversity

Even for a more classical and now perhaps a more mundane example, when I ask for recent information about “jaguars”, do I mean the car, the animal or the NFL football team? I probably only want one of these. When I make my request clear, a search engine should avoid ambiguous meanings.

Should a search engine remember and use my previous history of requests to better identify the most relevant documents personalized for me?

While the use of previous searches poses ethical and perhaps legal issues, using current interest in a topic, and geographical information does not seem intrusive. For example, in searching for “the best Italian restaurant”, we probably only want to consider restaurants “close” to me and in evaluating restaurants we probably only want to consider recent reviews. How much do we trust these reviews? However, once again maintaining a record of simple “best restaurant” searches does indicate where you have been.

The basic bag of words model

Suppose $\mathcal{C} = \{D_i\}$ is a collection of web documents (URLs).

- We can treat each document as a *bag of words*. Let's just say 200 words per document as some very rough average.
- Each query can also be considered as a very small bag of words. Most queries are two or three words. One estimate is an average of 2.2 words per query.
- The most naive approach. Find all the documents that contain all the words in the query. As a naive first approach call these the “relevant documents”.
- The most naive way to find all these (potentially) relevant documents would look at each document and check if all the query words occur.
- Even if all the documents were stored locally (which is not possible), **what would be a rough estimate for the time to find all the relevant documents?**

A quick calculation

You can do a quick calculation: compute $|\mathcal{C}| \cdot \frac{\text{number words}}{\text{document}} \cdot \frac{\text{number words}}{\text{query}}$ and then divide by $\frac{\text{number comparisons}}{\text{second}}$ to estimate the time for naively looking for documents that contain all the query terms.

Let's say that we have approximately 5 billion URLs, 200 words per document, 2 words per query which naively would result in $2 \cdot (10)^{12}$ comparisons. And let's say $(10)^7$ comparisons per second. Then a query would take $2 \cdot (10)^5 = 200,000$ seconds.

OK I might have some miscalculations but clearly this is *not* "real time" and not even feasible.

Making search feasible

One simple idea but very useful idea is the following. When a search engine *crawls* the web to find documents, it indexes documents so that for each *term* (i.e., word and frequent 2 word and 3 word phrases it maintains a sorted list of documents that contain that term. We usually ignore common articles such as “the”, “an, etc.

A term may also represent a number of strongly related terms. For example, a match for “cook” might be satisfied by “cooking”.

Making search feasible

One simple idea but very useful idea is the following. When a search engine *crawls* the web to find documents, it indexes documents so that for each *term* (i.e., word and frequent 2 word and 3 word phrases it maintains a sorted list of documents that contain that term. We usually ignore common articles such as “the”, “an, etc.

A term may also represent a number of strongly related terms. For example, a match for “cook” might be satisfied by “cooking”. You can get spelling suggestions, or maybe get a partial match, and sometimes be told that no documents match your query or there are no good matches but still get some suggested matches.

What can happen often is that there may be too many documents matching the query terms. So as we already suggested we really need a ranked list of documents in which the “most relevant” documents are ranked highest.

The vector space model and ranking documents

Instead of simply matching for query terms, we want to account for the fact that the occurrence of certain terms are more important for relevance.

Gerald Salton's idea was that a document (and a query) are represented by a vector of weighted counts of words/terms. Here are some ways to weight the occurrences of terms in a document.

- 1 Count the number of occurrences of a query term in a document, and better yet normalize this count by the relative frequency of terms in "the corpus of documents". This normalized count is called *tf-idf* standing for term frequency-inverse document frequency. Terms that occur infrequently throughout the corpus but appear frequently in a document should be weighted more. Wikipedia quotes a 2015 study that states "83 % of text based recommender systems in digital libraries use tf-idf".
- 2 Terms that appear in the title of the document or the title of a section heading should be given higher weights.
- 3 Terms that appear in the *anchor text* are important.

The vector space model continued

The above ideas for weighting terms are independent of the user queries. In contrast, we could also give higher weights to terms that relate to an individual's interests (say as learned by previous searches).

There can be many other ways to weight terms say by using machine learning techniques.

Now once we adopt this vector space representation, we can measure the similarity of a document and a query by say the cosine of these vectors.

An additional idea (in addition to the term similarity of the document and the query) is to exploit the “popularity” of a document. Popularity of a document in Google was done using *page rank* which is basically a random walk on the graph defined by the hyperlinks. This leads to a stationary probability distribution (i.e., an equilibrium) on the vertices (i.e., the (relevant) documents). That is, the probability of being at a particular vertex becomes a measure of the popularity of a document. We might apply a popularity measure restricted to just documents already identified as reasonably relevant.

Some further comments on the history of search engines

Page rank was touted as an essential idea in the early days of Google search but not clear how much of a role it now plays.

At about the same time as page rank, Jon Kleinberg introduced another graph based popularity method called *hubs and authorities* which was used in IBM's search engine (which they never commercialized).

With regard to *td-idf* (now accepted as an important idea), I saw the following comment in a web post (Language Log)
<https://languagelog.idc.upenn.edu/nll/?p=27770>

“one of Marvin Minsky's students once told me that Minsky warned him ‘If you're counting higher than one, you're doing it wrong’. Still, Salton's students (like Mike Lesk and Donna Harman) kept the flame alive.”

Marvin Minsky is recognized as one of the pioneers of artificial intelligence.

Why is search so profitable?

Companies such as IBM and (initially) Microsoft did not try to commercialize search, not recognizing the profitability of search. Indeed, should one charge for information or should the business model be based on advertising? Or was it possible that search would not be profitable?

We now know that search has turned out to be extremely profitable for companies based on advertising. The main way that Google and other companies sell advertising for search has spawned major research in algorithm design and auction theory. We will say more later about auctions, game theory and mechanism design.

We can view the process of assigning queries to advertisers (say wanting to display an *ad* as an *online bipartite graph matching problem*).

When a query arrives it needs to be assigned to one (or more, depending on how many advertising slots will be displayed) ads.

The “adwords” assignment problem

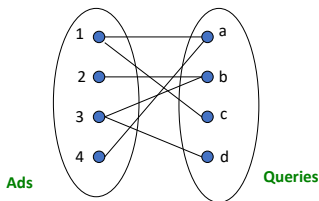


Figure: Figure taken from USC lecture notes by Rafael Ferreira da Silva

Each advertiser may have a budget (say for a given day) and indicates for given queries (or keywords) what it is willing to pay for that query but never exceeding its budget for all the queries assigned to that advertiser.

The search engine adjusts this advertiser's *bid* for a query based on how well it thinks the ad matches the query and then decides whether or not to assign an advertising slot to an advertiser and the price paid by the advertiser (depending on the slot) for each click by search users for the ad.

How does Facebook earn its revenue?

Aside: Later, we will discuss social networks as another killer application.

I assumed that Facebook and other social networks also generate most of their revenue by targeted ads. Indeed, I read a March 21, 2021 article that claims that about 98% of Facebook's revenue comes from targeted ads. But recently I read an October 23, 2021 newsletter by Heather Cox Richardson, a political historian, and Professor at Boston College. Her newsletters are widely circulated.

As you will see, she discusses information obtained from the Facebook whistleblower Frances Haugen. Professor Cox Richardson argues that Facebook does not make its money from selling ads. (I think this is intended to mean does not primarily make its money from targeted ads.).

I don't know enough about the revenue model for Facebook to know what fraction of its revenue Facebook derives from selling names to advertisers vs selling ads to users. But I do think this is a very interesting issue. The relevant parts of her newsletter appear on the next page. For emphasis, I have boldfaced what caught my attention.

Taken from Heather Cox Richardson newsletter

Today we learned that a Facebook researcher created a profile that appeared to be of a political conservative North Carolina mother and that within five days, Facebook's algorithm was steering the profile toward QAnon, a conspiracy theory touting then-president Trump as a secret warrior against a widespread pedophilia ring in the highest levels of government.

Although the fake profile did not follow those recommended groups, the profile was then inundated with groups and pages full of hate speech and disinformation. Other stories recently have emphasized that Facebook officials knew of the radicalization of users before the January 6 insurrection but declined to address the issue.

People often make the mistake of thinking that Facebook profits from the advertising it sells to users, but in fact the system works the opposite way. A media company profits from packaging users to sell to advertisers. Facebook has sliced and diced its users so that it can sell us with pinpoint accuracy to political interests eager to divide us or drive our votes.

The semantic web

We will end our discussion of search engines about where we began when I said, like other great ideas, sometimes these great ideas become so entrenched that it is hard to make further progress.

Is this the case with key word search? What kinds of “information needs” are beyond today’s search engines? See 2008 “Ontologies and the Semantic Web” article by Ian Horrocks and also his 2005 Lecture by the same title.

The vague goal of the semantic web is “to allow the vast range of web-accessible information and services to be more effectively exploited by both humans and automated tools.”

A more specific goal is to *integrate* information that occurs in the web but not in one document.

Some specific examples of information that might not exist in any one document

One example Horrocks gave is to retrieve a “list of all the heads of state of EU countries”. Of course, once such an example is given, it is likely (as in this example) that one can successfully find the required information in a single query. **Why was this a difficult search in 2008 and an easy search today? It was the fourth document in my search on October 17, 2021.**

“The classic example of a semantic web application is an automated travel agent that, given various constraints and preferences, would offer the user suitable travel or vacation suggestions”. This example still seems beyond something we can easily do with current search engines.

I decided to create the following query “list of all computer scientists whose last name is Cook”. In my first search, most of the retrieved documents are not useful but the first of the retrieved documents is for Stephen Cook and the second document is a very incomplete list of computer scientists.

Screenshot of my query for computer scientists with last name Cook

The screenshot shows a Google search results page for the query "list of computer scientists whose last name is Cook". The search bar contains the query, and the results are displayed below. The first result is from Encyclopedia.com, titled "Stephen Arthur Cook | Encyclopedia.com". The second result is from TheBestSchools.org, titled "The Most Influential Computer Scientists - TheBestSchools.org". Below the search results is a section titled "People also search for" with three suggestions: "leonid levin famous computer scientists and their inventions", "gordon cook 10 computer inventors and their inventions", and "stephen cook obituary 20 computer inventors and their inventions". The third result is from Future Students | York University, titled "Computer Science | Future Students | York University". The fourth result is from lamturing.acm.org, titled "Stephen A Cook - A.M. Turing Award Laureate". The fifth result is from books.google.ca, titled "Coding as a Playground: Programming and Computational ...". The sixth result is from esuonline.asu.edu, titled "Online master of computer science (MCS)".

Chrome File Edit View History Bookmarks Profiles Tab Window Help

Inbox (15,346) - abborndi x New Tab x G list of computer scientist: x G list of all the heads of sta: x +

google.com/search?q=list+of+computer+scientists+whose+last+name+is+Cook&rlz=1C8CHFA_enCA904CA... Update

Apps Getting Started Latest Headlines Imported From Fir... CSC200_Lecture4... Application List, D... Reading List

Google list of computer scientists whose last name is Cook X

https://www.encyclopedia.com/science/stephen-arth...
Stephen Arthur Cook | Encyclopedia.com
He earned his M.S. from Harvard in 1962, and his Ph.D. in 1966. He then took a position as assistant professor of mathematics and computer science at the ...
Missing: name | Must include: name

https://thebestschools.org/magazine/most-influential...
The Most Influential Computer Scientists - TheBestSchools.org
Sep. 7, 2021 — Who are the scientists shaping and framing the computer-driven world ... to put names and faces to the esoteric acronyms and the machinery.

People also search for

leonid levin	famous computer scientists and their inventions
gordon cook	10 computer inventors and their inventions
stephen cook obituary	20 computer inventors and their inventions

https://futurestudents.yorku.ca/program/computer-s-...
Computer Science | Future Students | York University
This program is intensive in Mathematics and Computer Science courses. ... am a high-school student I have completed at least one year of full-time study at ...

https://amturing.acm.org/cook_n991950
Stephen A Cook - A.M. Turing Award Laureate
Cook entered the University of Michigan in 1957, majoring in science engineering. He was introduced to computer programming in a freshman course taught by ...
Missing: name | Must include: name

https://books.google.ca/books
Coding as a Playground: Programming and Computational ...
Marina Umaschi Bers · 2020 · Education
Basic research must inform the debate about the role of computer science in the ... and looking up a name in an alphabetical list (linear; starting at the ...

https://esuonline.asu.edu/.../Online graduate programs
Online master of computer science (MCS)

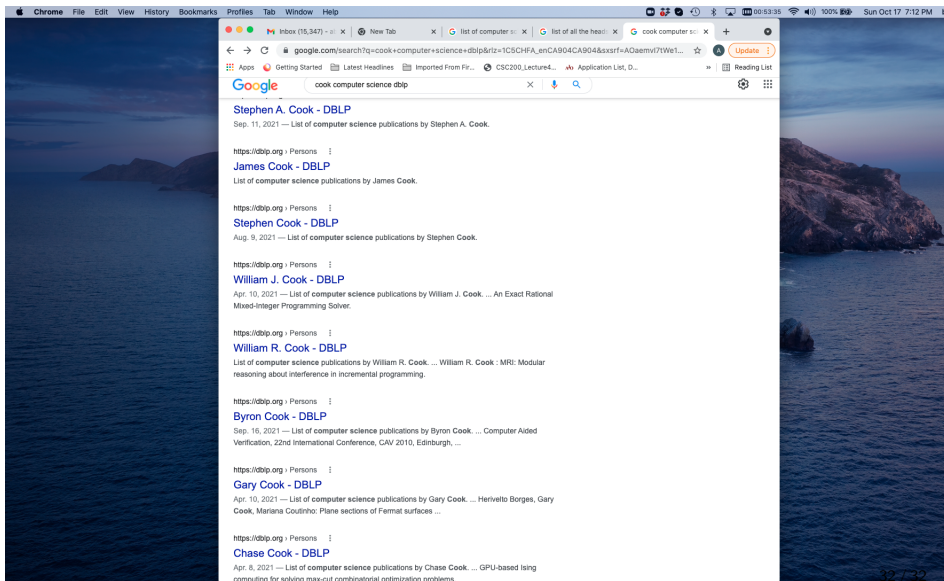
My second attempt to find a list of computer scientists with last name Cook

One thing that key word search has somehow taught us is to modify our searches. I know that published papers in computer science have a good chance of being listed on a “dblp” site.

So I created a search with the query “cook computer science dblp”

Now the documents returned by this query is more informative. Of course, I assume this is not at all an exhaustive list of computer scientists named Cook.

Another search to find other computer scientists with last name Cook



The screenshot shows a Google search results page for the query "cook computer science dblp". The search results are listed in descending order of date. Each result includes a link to a DBLP profile, the name of the person, and a brief description of their work.

Search results for "cook computer science dblp":

- <https://dblp.org> > Persons > **Stephen A. Cook - DBLP**
Sep. 11, 2021 — List of computer science publications by Stephen A. Cook.
- <https://dblp.org> > Persons > **James Cook - DBLP**
List of computer science publications by James Cook.
- <https://dblp.org> > Persons > **Stephen Cook - DBLP**
Aug. 9, 2021 — List of computer science publications by Stephen Cook.
- <https://dblp.org> > Persons > **William J. Cook - DBLP**
Apr. 10, 2021 — List of computer science publications by William J. Cook. ... An Exact Rational Mixed-Integer Programming Solver.
- <https://dblp.org> > Persons > **William R. Cook - DBLP**
List of computer science publications by William R. Cook. ... William R. Cook : MRI: Modular reasoning about interference in incremental programming.
- <https://dblp.org> > Persons > **Byron Cook - DBLP**
Sep. 16, 2021 — List of computer science publications by Byron Cook. ... Computer Aided Verification, 22nd International Conference, CAV 2010, Edinburgh, ...
- <https://dblp.org> > Persons > **Gary Cook - DBLP**
Apr. 10, 2021 — List of computer science publications by Gary Cook. ... Herivelto Borges, Gary Cook, Mariana Coutinho: Plane sections of Fermat surfaces ...
- <https://dblp.org> > Persons > **Chase Cook - DBLP**
Apr. 8, 2021 — List of computer science publications by Chase Cook. ... GPU-based Ising computing for solving max-cut combinatorial optimization problems.