Algorithms find solutions, so what’s the problem?

Mathijs de Weerdt, June 7, 2023
But we aren’t using algorithms for finding the solution to some very relevant problems.
Outline

1. What is an algorithm?
2. What is the problem?
3. How can we address this?
algorithm = recipe to be executed by a computer
Dijkstra’s algorithm (1959)

“The question of whether a computer can think is no more interesting than the question of whether a submarine can swim.”

— Edsger W. Dijkstra
function Dijkstra$(V,E)$

- $d[v] \leftarrow \infty$ for all $v \in V$
- $d[s] \leftarrow 0$ for start vertex
- while $V$ is not empty do
  - $m \leftarrow \arg\min_{v \in V} d[v]$
  - Remove $m$ from $V$
  - if $d[m] = \infty$ then
    return $\infty$
  - for every $e \leftarrow (m,u) \in E$ do
    if $d[m] + w(e) < d[u]$ then
      $d[u] \leftarrow d[m] + w(e)$

return $d$
function Dijkstra(V, E)
    d[v] ← ∞ for all v ∈ V
    d[s] ← 0 for start vertex
    while V is not empty do
        m ← arg min d[v]  
            v ∈ V
        Remove m from V
        if d[m] = ∞ then
            return ∞
        for every e ← (m, u) ∈ E do
            if d[m] + w(e) < d[u] then
                d[u] ← d[m] + w(e)
        return d
function Dijkstra(V, E)
    d[v] ← ∞ for all v ∈ V
    d[s] ← 0 for start vertex
    while V is not empty do
        m ← arg min_{v ∈ V} d[v]
        Remove m from V
        if d[m] = ∞ then
            return ∞
        for every e ← (m, u) ∈ E do
            if d[m] + w(e) < d[u] then
                d[u] ← d[m] + w(e)
        return d

http://www.danielkoitzsch.de/blog/2015/11/06/visual-comparison-of-dijkstra-and-a-star/
function Dijkstra(V, E)
    d[v] ← ∞ for all v ∈ V
    d[s] ← 0 for start vertex
    while V is not empty do
        m ← arg min _v∈V d[v]
        Remove m from V
        if d[m] = ∞ then
            return ∞
        for every e ← (m, u) ∈ E do
            if d[m] + w(e) < d[u] then
                d[u] ← d[m] + w(e)
        return d

http://www.danielkoitzsch.de/blog/2015/11/06/visual-comparison-of-dijkstra-and-a-star/
function Dijkstra(V, E)
    d[v] ← ∞ for all v ∈ V
    d[s] ← 0 for start vertex
    while V is not empty do
        m ← arg min d[v]  
            v∈V
        Remove m from V
        if d[m] = ∞ then
            return ∞
        for every e ← (m, u) ∈ E do
            if d[m] + w(e) < d[u] then
                d[u] ← d[m] + w(e)
        return d
Data = ingredients

▪ structured (e.g. a map)
▪ unstructured:
  ▪ patterns trained and used (machine learning)
  ▪ collected by the algorithm itself (reinforcement learning)
An example with a *lot of* data: large language models

e.g. ChatGPT: GPT-4 was trained on 570 gigabytes of data

570 000 000 000 bytes

(characters)

thick book
An example with a lot of data: large language models

“all” data on the internet estimated to be 5 billion gigabytes (by former CEO of Google, Eric Schmidt, interview in 2023)

5 000 000 000 000 000 000 000 000

bytes
(characters)

thick book

movie

https://www.easytechjunkie.com/how-big-is-the-internet.htm
The quality of the ingredients counts!
Wind farm wake control


https://github.com/AlgTUDelft/wind-farm-env
Algorithms should be used for problems that matter!

Many of these are **planning or scheduling** problems!

- train shunting/path finding
- energy system investments
- heat network control
- efficient (bio)manufacturing
Planning & Scheduling Methods in Practice

Surprisingly little use of such algorithmic insights in practice!

- “the majority of derived mathematical solutions have not been used in practice by manufacturing businesses” (Jackson et al., 2004)
- “Spreadsheet application still dominates enterprise resource planning and advanced planning systems” (De Man, Strandhagen, 2018)
- “most schedulers prefer the use of a simulation-based software or manual decision, which result to suboptimal solutions” (Georgiadis et al., 2019)

So what’s the problem?
Traveling Salesman

find the shortest route via 
a number of locations
Traveling Salesman Package delivery

find the shortest route via a number of locations
Problem complexity

Time needed for the computation (runtime)

Input size of the problem: $n$

$2^n$
P (easy)

P is the class of problems that can be solved in polynomial time.

Examples

▪ Shortest path
▪ Scheduling with unit processing times

NP (-hard)

NP-hard are those problems for which we have no polynomial-time algorithms.

Examples

▪ Traveling salesman problem
▪ Scheduling with arbitrary processing times, release times and deadlines
So what now?

• algorithmic innovations (possibly not optimal), or
• don’t solve the whole problem…
Choosing the right model:

https://commons.wikimedia.org/wiki/File:HPMOR_fan_art.jpg
https://www.cgtrader.com/3d-models/character/man/lego-harry-potter
https://rigmodels.com/model.php?view=Harry-Potter-3d-model__REQ0I1Y8QAPMNSVVG3ALT7WF5
Choosing the right model requires...

algorithmic understanding
1. what is computational challenging?
2. which algorithm to use?

and expertise on the domain
1. what’s important for (in)feasibility?
2. what can we fix later?
How to address this?

In my own research

Supporting others

EnergySHR
Shunting at railway hubs

https://nieuws.ns.nl/media-archief/
Shunting at railway hubs

Eindhoven
Using a more simple model can help!

Example: RAIL lab – LPT Robust (2023-2027)

ICAI Lab, Scientific directors: Marjan van den Akker (UU), Mathijs de Weerdt (TU Delft)

Five PhD projects:
1. Cooperation between human and AI planners
2. Robust planning
3. Quickly reacting to changes and disruptions
4. Supporting strategic decisions regarding the infrastructure capacity
5. Learning from previous situations and produce recognizable plans
The energy transition

- not completely controllable,
- variable,
- not perfectly predictable
• need for energy storage
• demand management
• more active role for end users
• new business models and markets
Scheduling what to bid & when to charge


https://github.com/AlgTUDelft/B-FELSA
Optimal Investments in and Operation of the Energy System

- whole of Europe
- 20+ million constraints
- no optimal solution after a week
- just for the power system…
Work in a team with algorithmic & domain expertise

Core aims:

- also **use algorithms to find models** with a good trade-off between accuracy and computation
- and to allow **making trade-offs interactively with the user**
Other projects

https://github.com/ftbv/grid-penguin

by Jichen Wu

Kim van der Houten

Ksenija Stepanovic

Grigori Veviurko

Junhan Wen
Algorithmic innovations

Domain researchers / experts

shared data, models, simulators, solutions

Guidelines for modelling

Understanding when to use which algorithm

Practitioners, users
How to address this?

In my own research

Supporting others

EnergySHR
Center for Energy System Intelligence
co-chaired with Yashar Ghiassi-Farrokhfal (RSM)

EnergySHR
Sharing data and code by and for researchers
- For practitioners: to benefit more directly from new algorithms
- For researchers: working with real data; solving the right problems

Contact me (or Caroline Duterloo)!

Inspired by The Green Village: a living lab for sustainability and energy research

https://www.thegreenvillage.org
NL AIC

- transfer of knowledge
  https://energie.ai-cursus.nl/home
- bringing together problem owners and AI expertise (researchers & developers)

Contact me if you’re looking for AI expertise!
Students are all different

Teach bachelor students about research

More than 250 papers in less than 2 months

A big change

But they did: the 201st studies are full of interesting results, from programmes that enable a computer to defend itself better in a game of AI football to algorithms that automatically analyse web content. "Surprisingly, including professors, were more than positively surprised by the quality," says Mathijs enthusiastically. "Some are even working on getting their research published."

An unexpected puzzle

For Mathijs and his team there was also a major task: how do you ensure that all these students set up a good research project?
Education: the bachelor research project

Challenge:
- experience research
- write a paper individually
- over 300 students per year, 60-90 projects

Next:
- connect CS students (more) to other disciplines
- allow other students to learn (more) about algorithms

[Table of Project Popularity]

https://projectforum.tudelft.nl/

with Philipe Louchtch, Taico Aerts, Gosia Migut
Conclusion

Algorithmic insights *can* help solve society’s problems.

But it is important that an algorithm
1. uses the right model
2. uses the right data
3. works in interaction with users
Not alone

- My promotors: great to have you here!
- PhD students: current and former
- Post-docs, research software engineers
- Business developers, project support, etc.

Konstantin Sidorov  Ksenija Stepanovic  Issa Hanou  Maaike Elgersma  Koos van der Linden  Kim van der Houten  Grigorii Veviurko  Junhan Wen
Not alone

• Other staff of the section
  Algorithmics
• All colleagues of the two
departments
• My collaborators over the years
• Industry partners
• Could not mention all in this talk
• Fantastic to work with you!

TU Delft