FIELDS INSTITUTE
FOR RESEARCH IN MATHEMATICAL SCIENCES

# Workshop on the Use of Quantum Algorithms/Formalisms in Finance

## Quantum Optimization: What (not) to Expect?

Stefan Creemers

October 16, 2025

# Optimization 101

- Goal = maximize (or minimize) some objective function given a number of constraints.
- More formally:

  $\max V(\boldsymbol{x})$

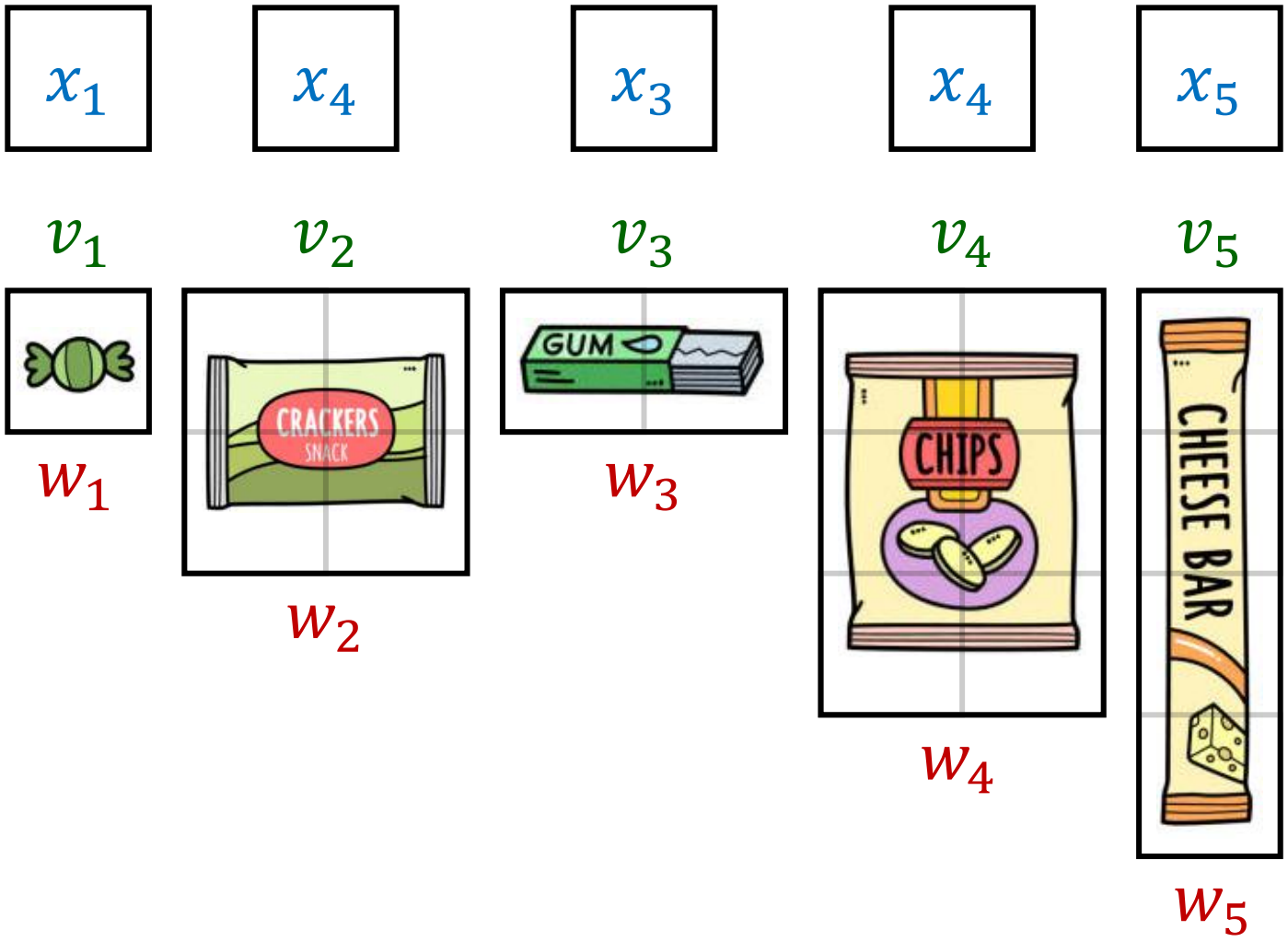
  subject to

  $\boldsymbol{x} \in \Omega$
- Where:
  - $V(\boldsymbol{x})$ is the objective function that returns the value of solution $\boldsymbol{x}$.
  - $\boldsymbol{x} = \{x_1, x_2, \dots, x_n\}$ is a solution that has $n$ decision variables.
  - $x_i$ is the value assigned to decision variable $i$.
  - $\Omega$ is the set of all feasible solutions that do not violate the problem constraints.
- Typical example of an optimization problem: the knapsack problem.

# Knapsack problem

$$\max \sum_{i=1}^{n} v_i x_i$$

$$subject\ to: \sum_{i=1}^{n} w_i x_i \leq W$$



$W = 12$

# Knapsack problem

$$\max \sum_{i=1}^{n} v_i x_i$$

$$subject\ to: \sum_{i=1}^{n} w_i x_i \leq W$$

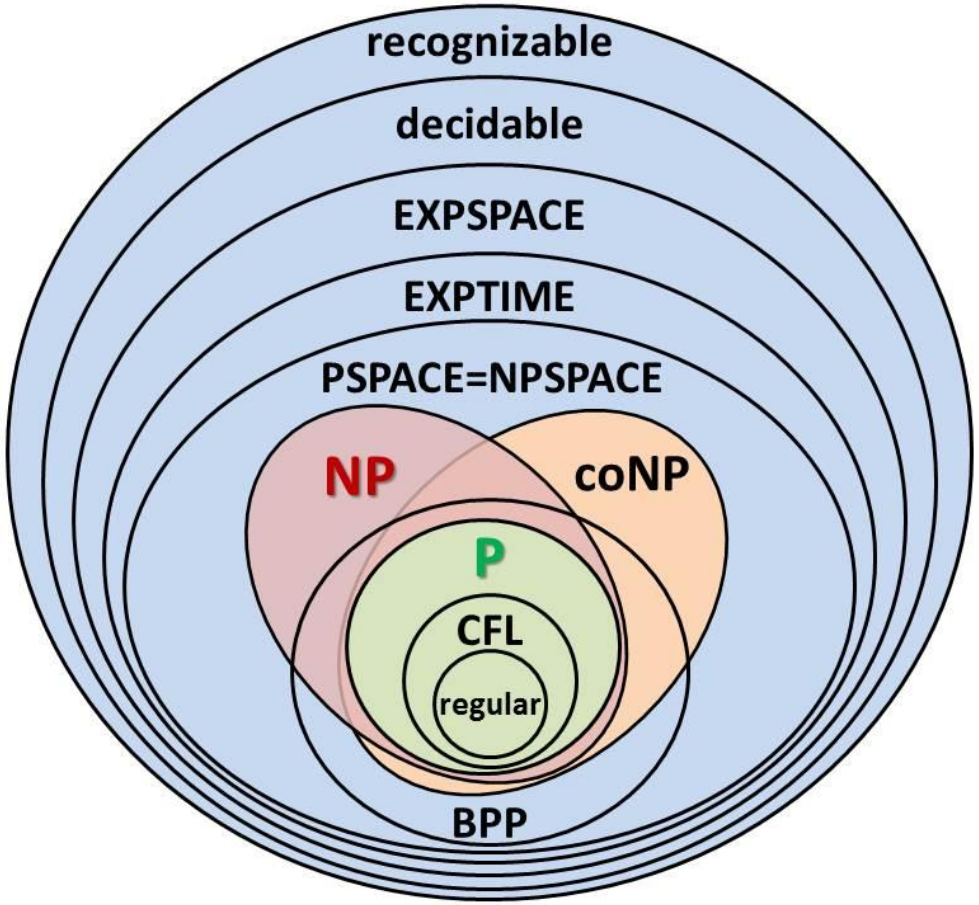
| 1 | 1 | 1 | 0 | 1 |
|---|---|---|---|---|
| 2€ | 6€ | 4€ | 2€ | 8€ |
| 1 | 4 | 2 | 6 | 4 |

$$V(\boldsymbol{x}) = 20€$$

# Lots of (optimization) problems…

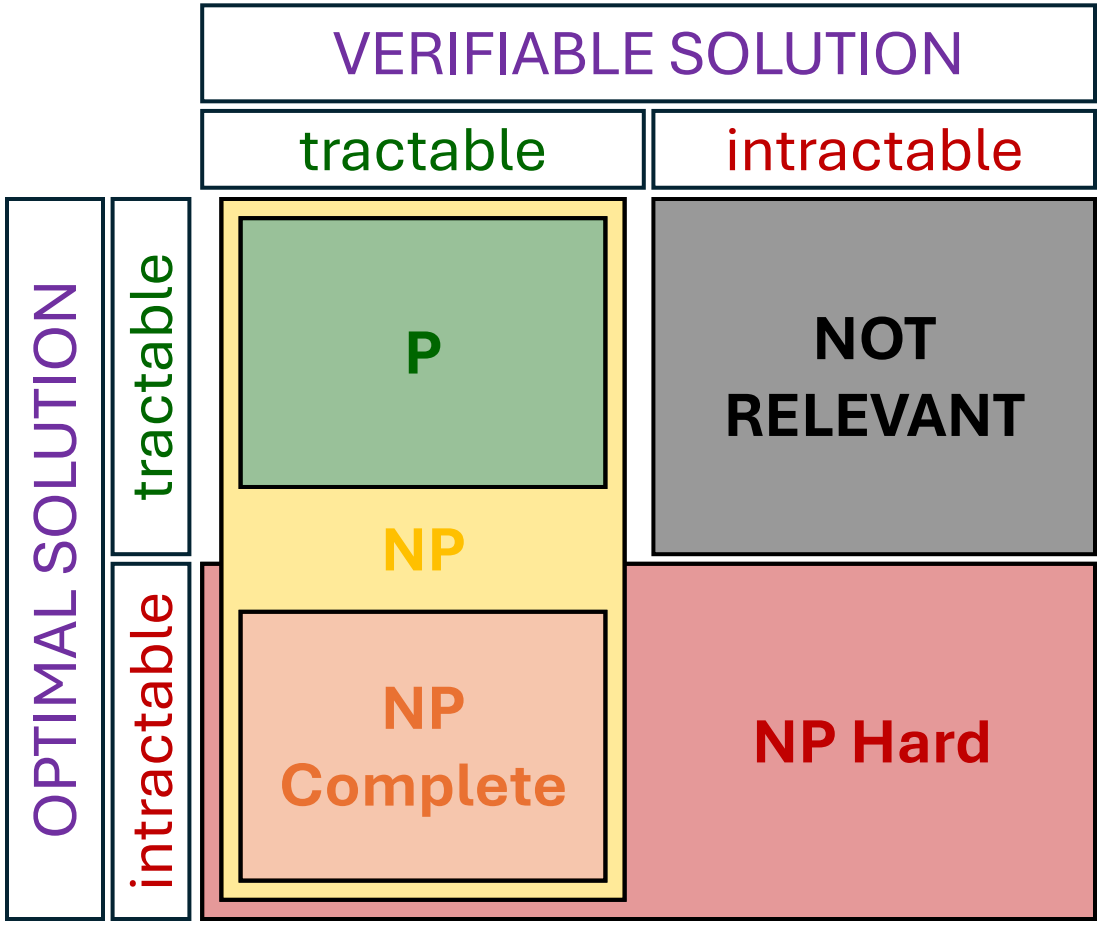
➔ lots of problem classes…

- Problems can be classified based on their "computational complexity".
- A number of complexity classes are or particular interest:
  - **P**
  - **NP**
  - **NP Complete**
  - **NP Hard**

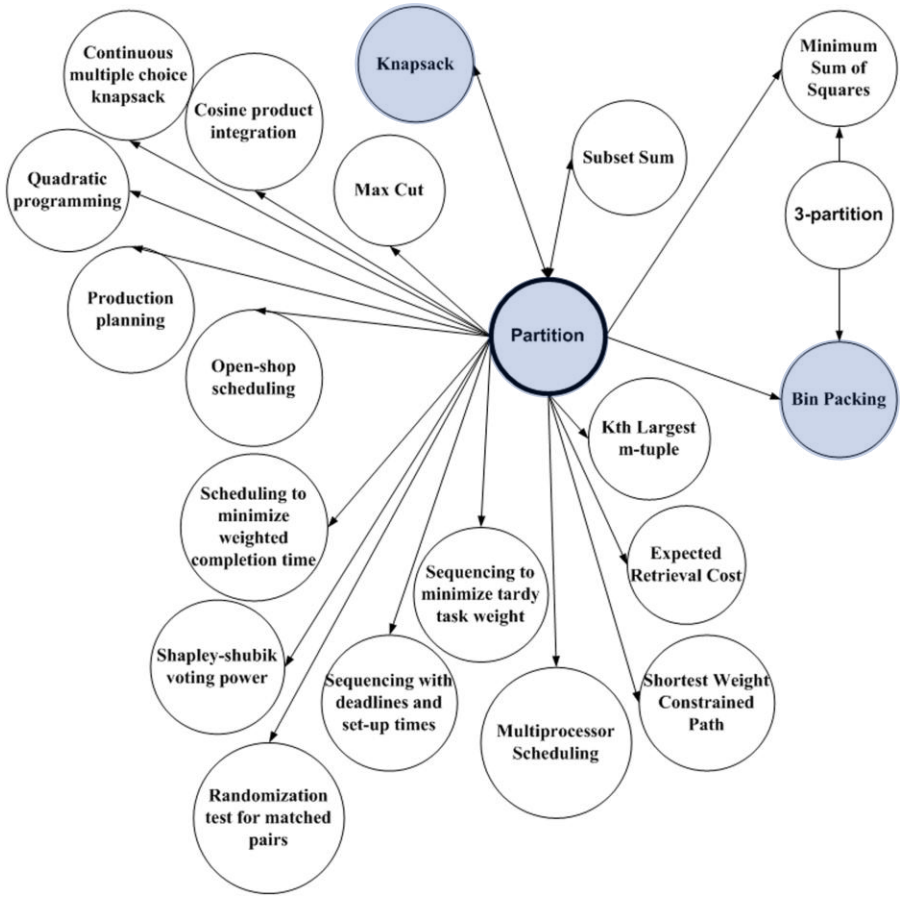


# Computational complexity 101

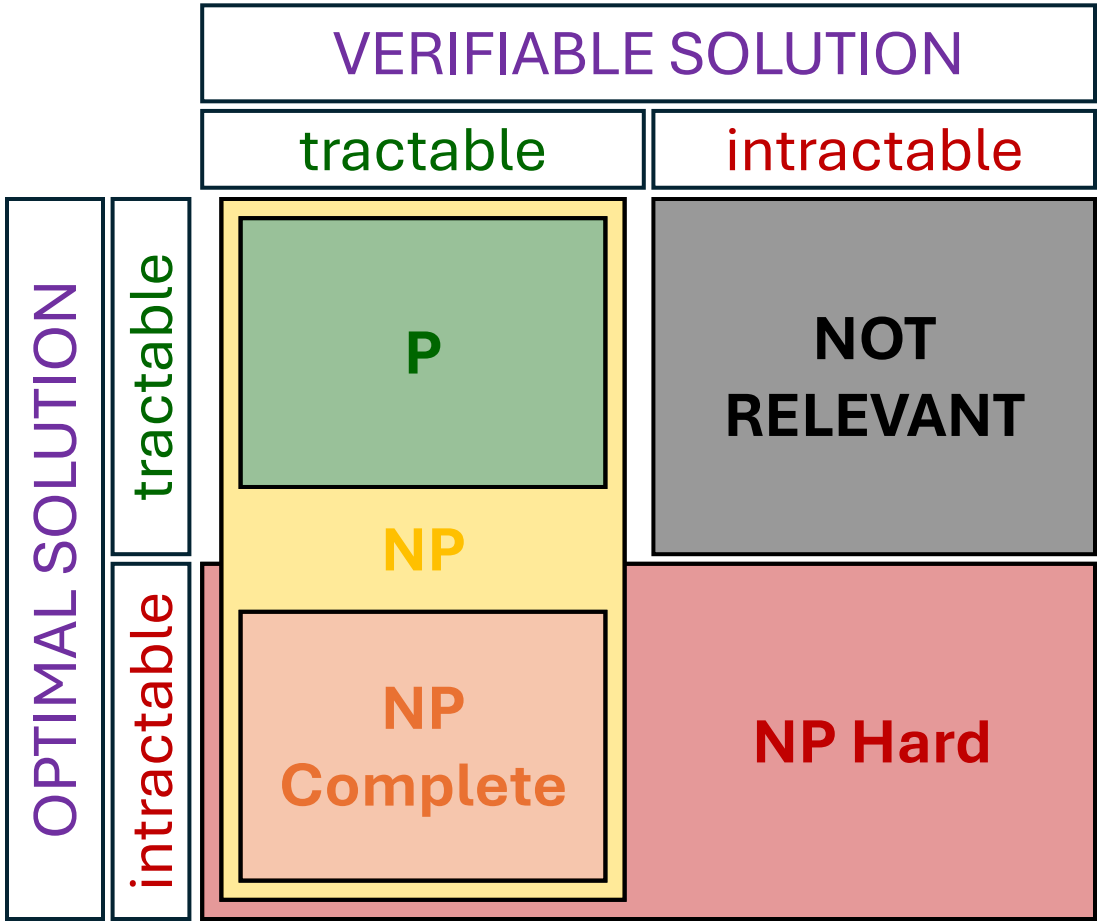
- **P**, **NP**, **NP Complete**, and **NP hard** differ with respect to:
  - Whether a solution is verifiable in polynomial time.
  - Whether an optimal solution can be obtained in polynomial time.
- Polynomial time is considered tractable and requires a less-than exponential running time (e.g., $O(n)$, $O(n^2)$, $O(n^k)$...).
- Many optimization problems are NP-Complete/Hard and are considered intractable (i.e., they require running time $O(2^n)$, $O(n!)$, or worse).
- This diagram assumes P ≠ NP.



# Example NP-complete optimization problems

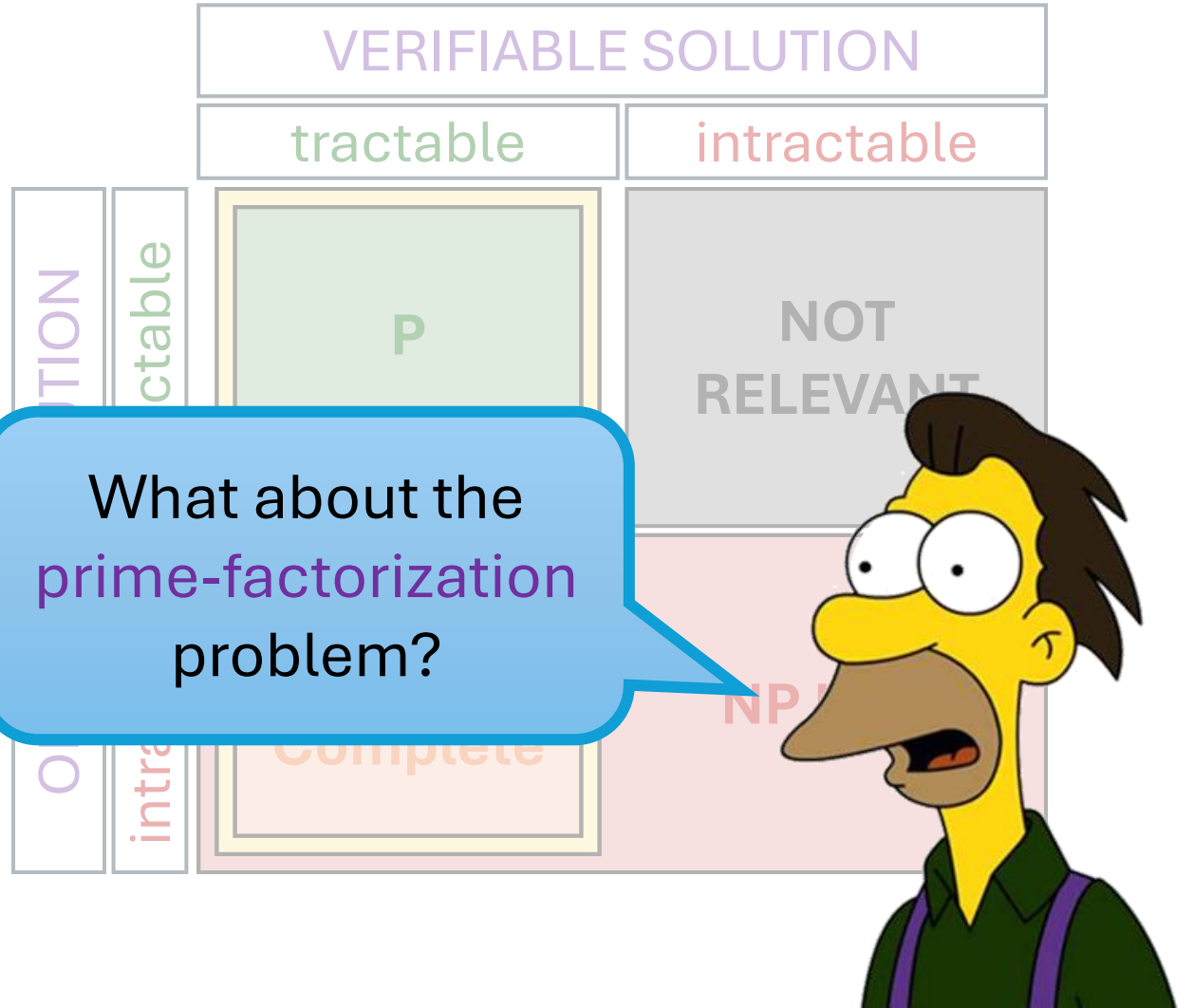



Ruiz-Vanoye et al. (2011). *Survey of polynomial transformations between NP-complete problems*. Journal of Computational and Applied Mathematics.



# Example NP-complete optimization problems

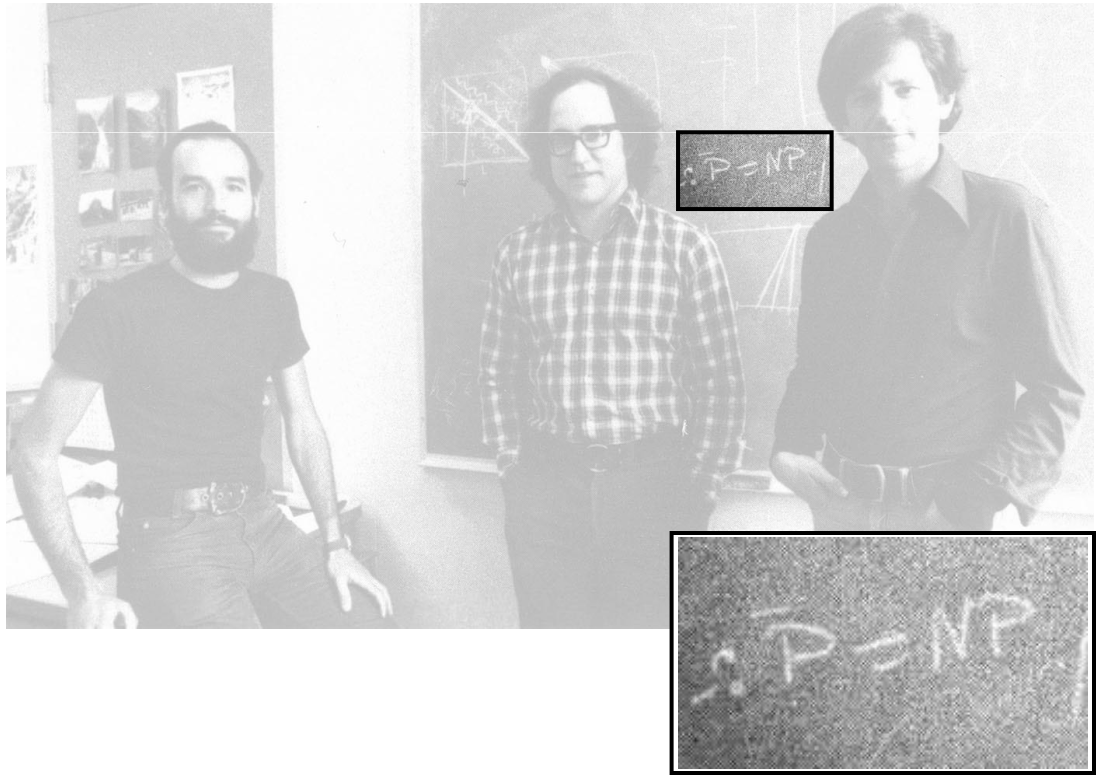
Ruiz-Vanoye et al. (2011). *Survey of polynomial transformations between NP-complete problems*. Journal of Computational and Applied Mathematics.



# Prime-factorization problem

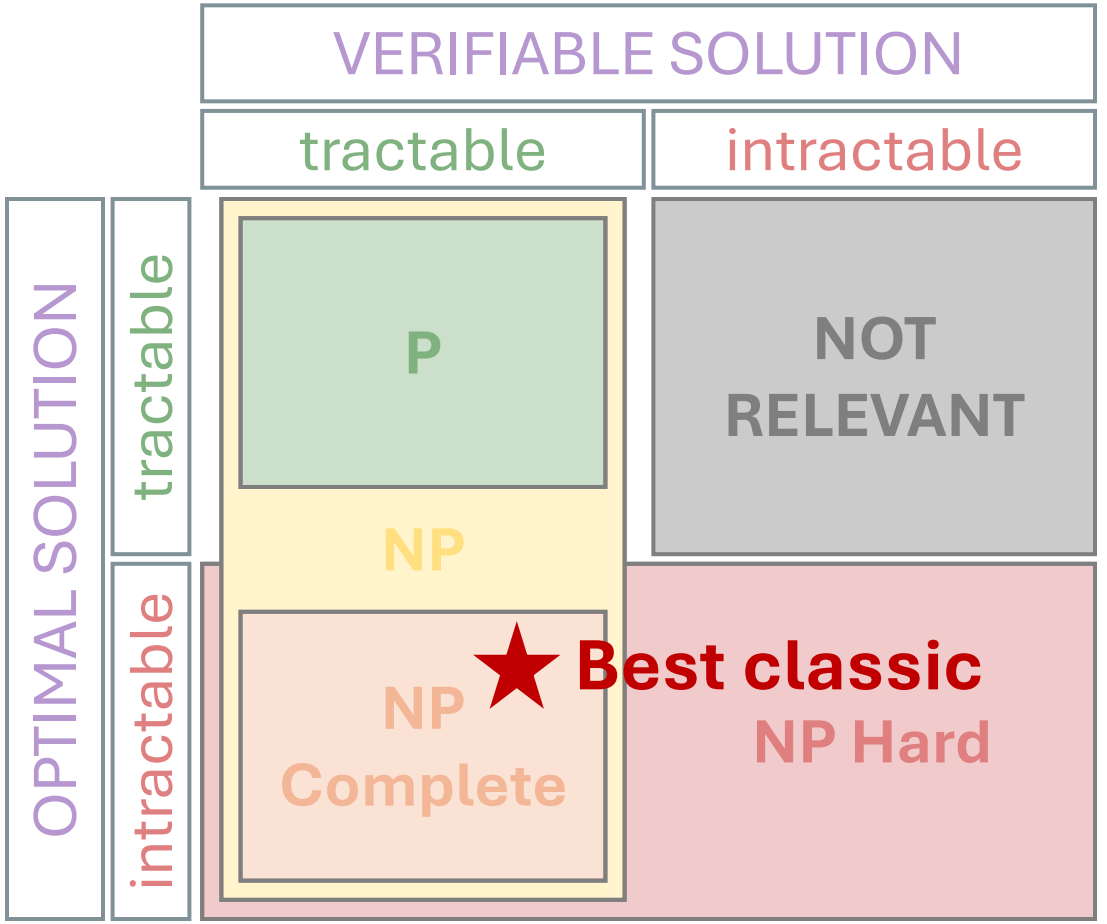
- Most encryption schemes that are used today rely on large primes (e.g., RSA).
- If primes can be factorized efficiently, these encryption schemes can be broken in polynomial time.
- The best-known classical algorithm to factorize primes runs in (sub-)exponential time…

- Rivest, Shamir, & Adleman (RSA)



# Prime-factorization problem

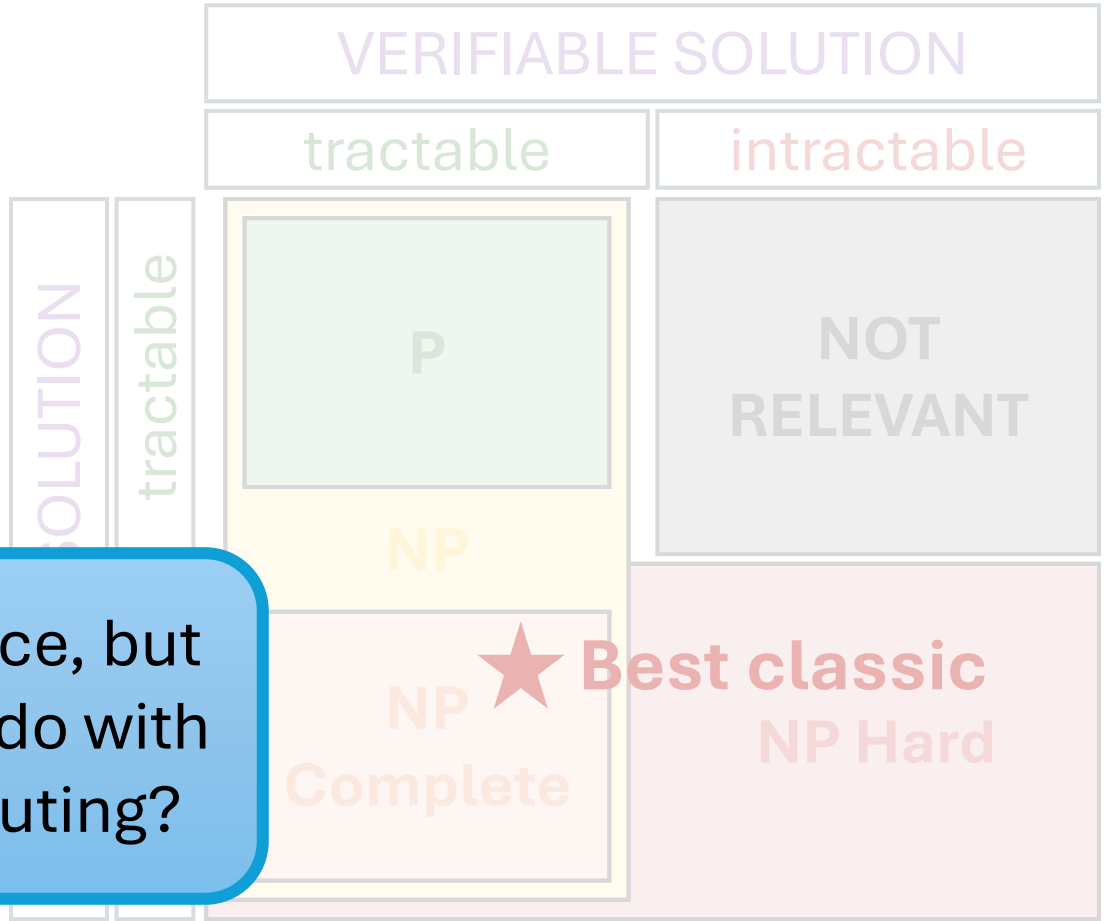
- Most encryption schemes that are used today rely on large primes (e.g., RSA).
- If primes can be factorized efficiently, these encryption schemes can be broken in polynomial time.
- The best-known classical algorithm to factorize primes runs in (sub-)exponential time…



# Prime-factorization problem

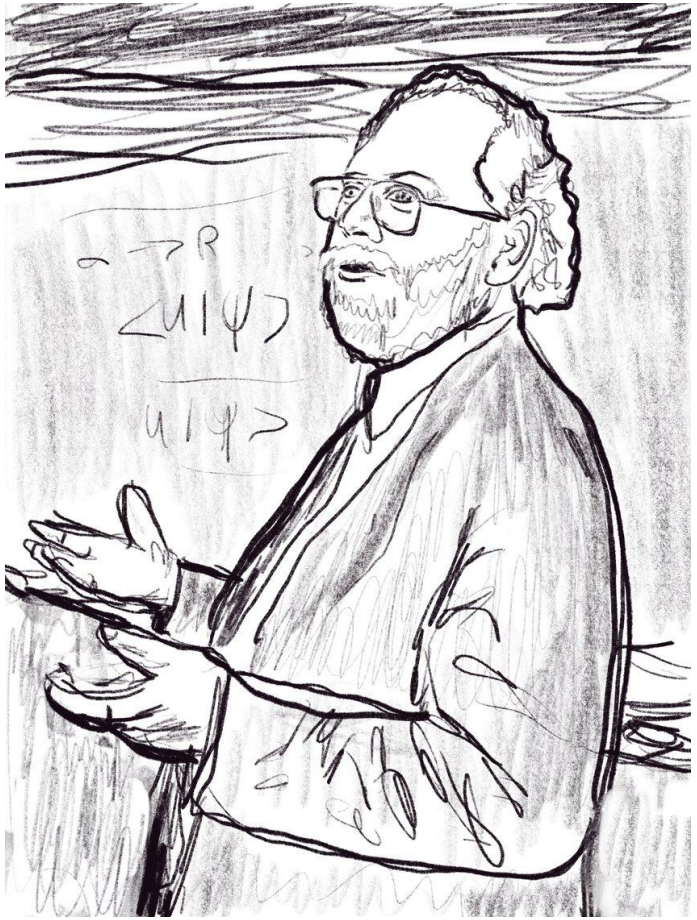
- Most encryption schemes that are used today rely on large primes (e.g., RSA).
- If primes can be factorized efficiently, these encryption schemes can be broken in polynomial time.
- The best-known algorithm to factorize runs in (sub-)exponential time.

This is all very nice, but what has this to do with quantum computing?



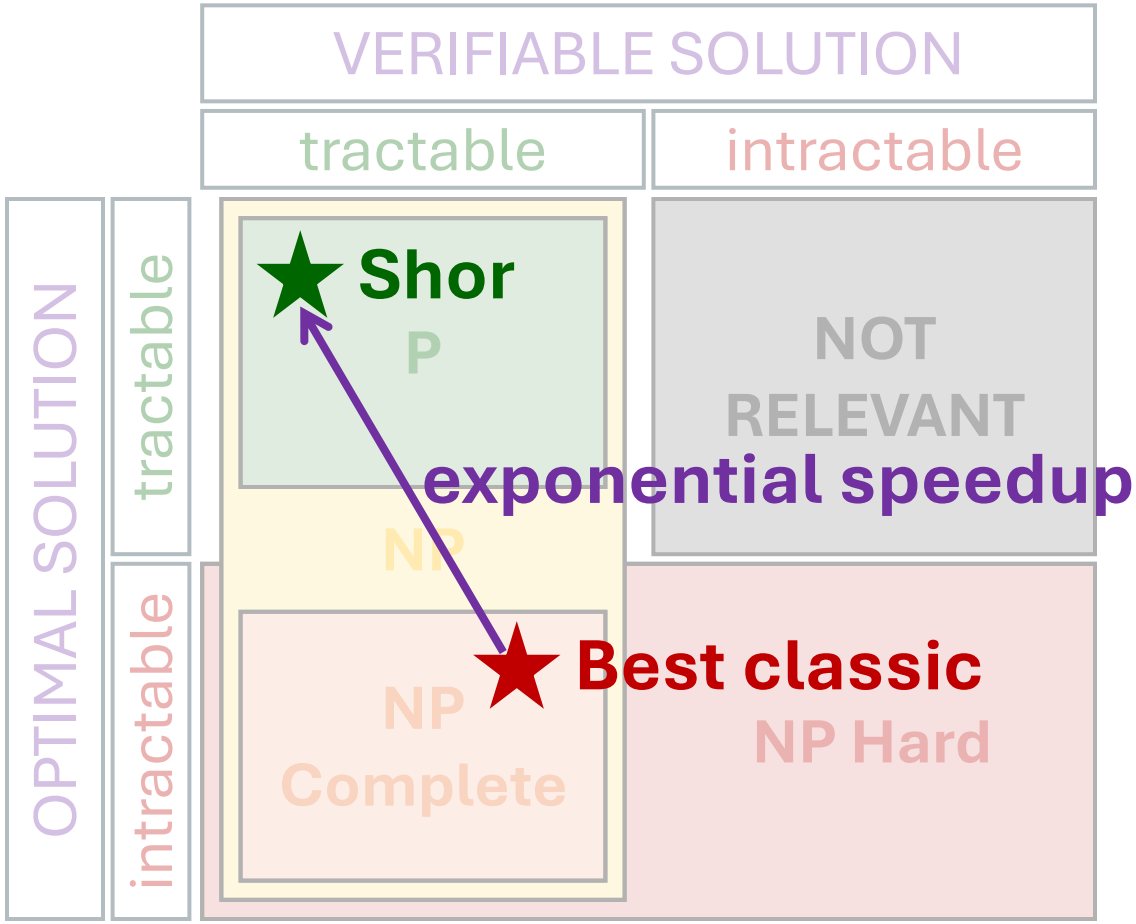
# Prime-factorization problem

- In 1994 Shor published a quantum algorithm that can factorize primes in polynomial time, resulting in an exponential speedup!



# Prime-factorization problem

- In 1994 Shor published a quantum algorithm that can factorize primes in polynomial time, resulting in an exponential speedup!
- Does this imply we can get an exponential speedup also when solving NP-complete problems? Can we solve NP-complete problems in polynomial time?
- Unfortunately, no, because it has never been shown that the prime-factorization problem is NP-complete… In fact, there may very well be a classical algorithm that can solve the prime-factorization problem in polynomial time.

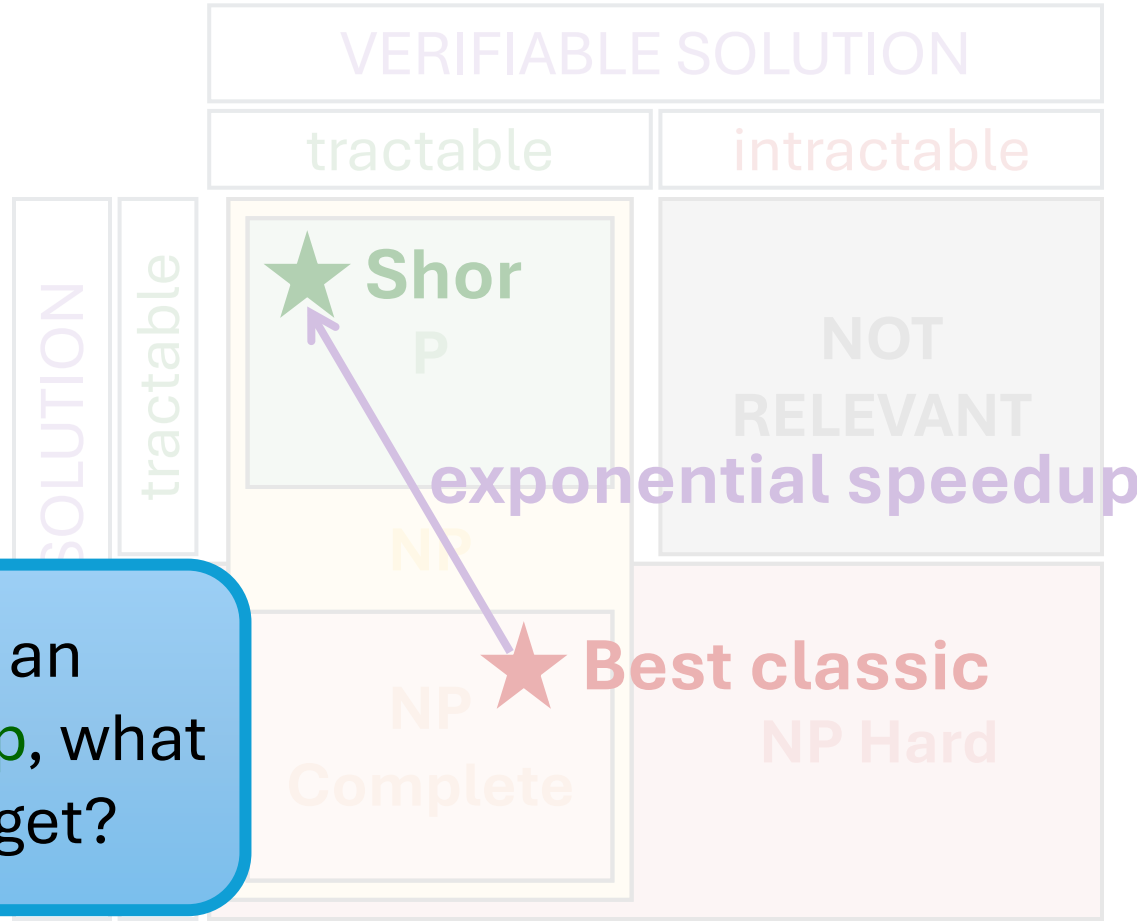


# Prime-factorization problem

- In 1994 Shor published a quantum algorithm that can factorize primes in polynomial time, resulting in an exponential speedup!
- Does this imply we can get an exponential speedup also when solving NP-complete problems? Can we solve NP-complete problems in polynomial time?

Unfortunately, no, because it has never been shown that there is a classical algorithm for factorization in polynomial time

If we cannot get an exponential speedup, what speedup can we get?



# A true Lov story

- Imagine yesterday Carl went out yesterday and he met the girl of his dreams!
- It gets even better: he got her phone number!
- Unfortunately, however, Carl completely forgot the name of this dream girl…

# A true Lov story

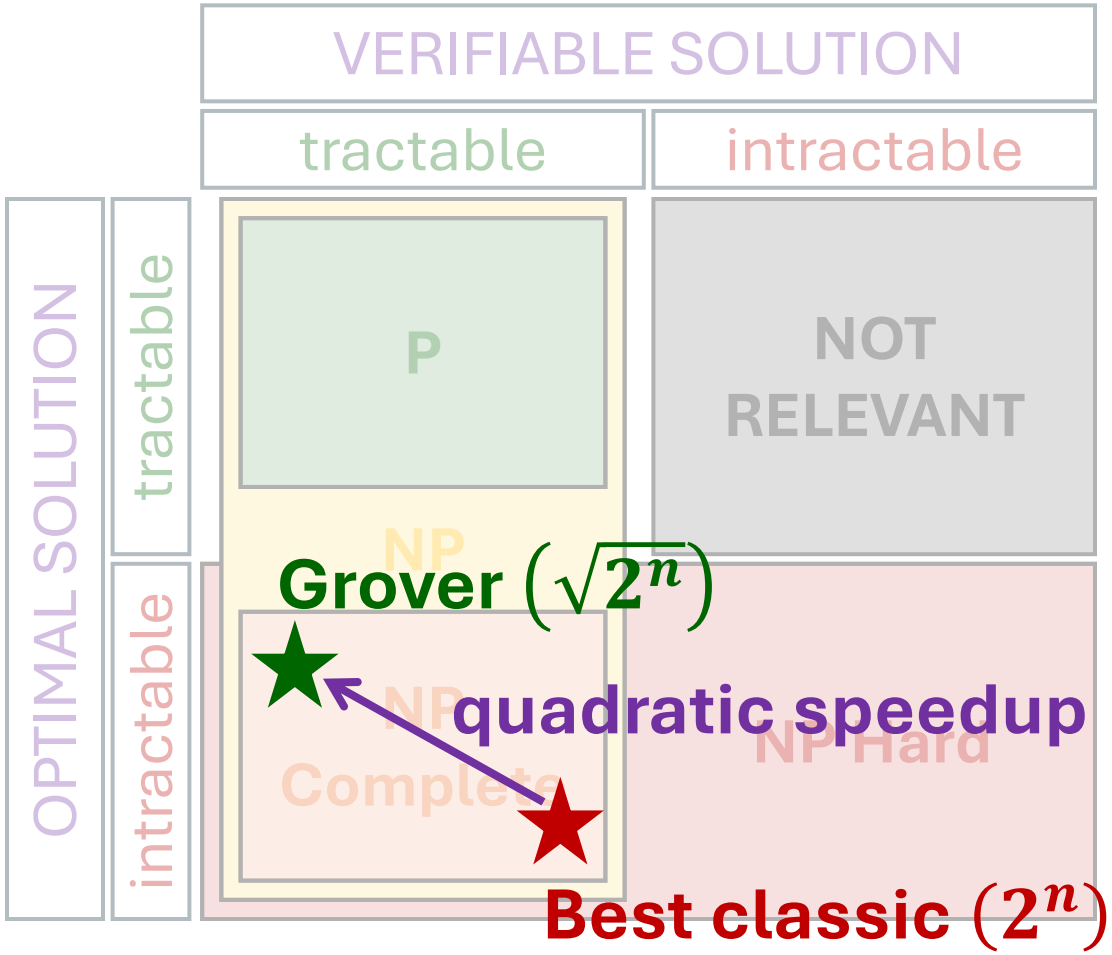
- Imagine yesterday Carl went out yesterday and he met the girl of his dreams!
- It gets even better: he got her phone number!
- Unfortunately, however, Carl completely forgot the name of this dream girl…
- Luckily, Carl has a phone book!
- The phone book, however, is ordered by name, not by phone number.
- As Carl is determined to retrieve the name of his crush, he starts browsing the phone book.
- As there are $2^n$ entries in the phone book, in the worst-case, Carl will have to check $2^n$ phone numbers. This might take a while…

# Lov Grover to the rescue!

- In 1996, Grover published an algorithm that can find a target entry in an unstructured database that has $2^n$ entries by looking at only $\sqrt{2^n}$ entries.
- Compared to a classical approach (that requires to look at all $2^n$ entries), this results in a quadratic speedup!

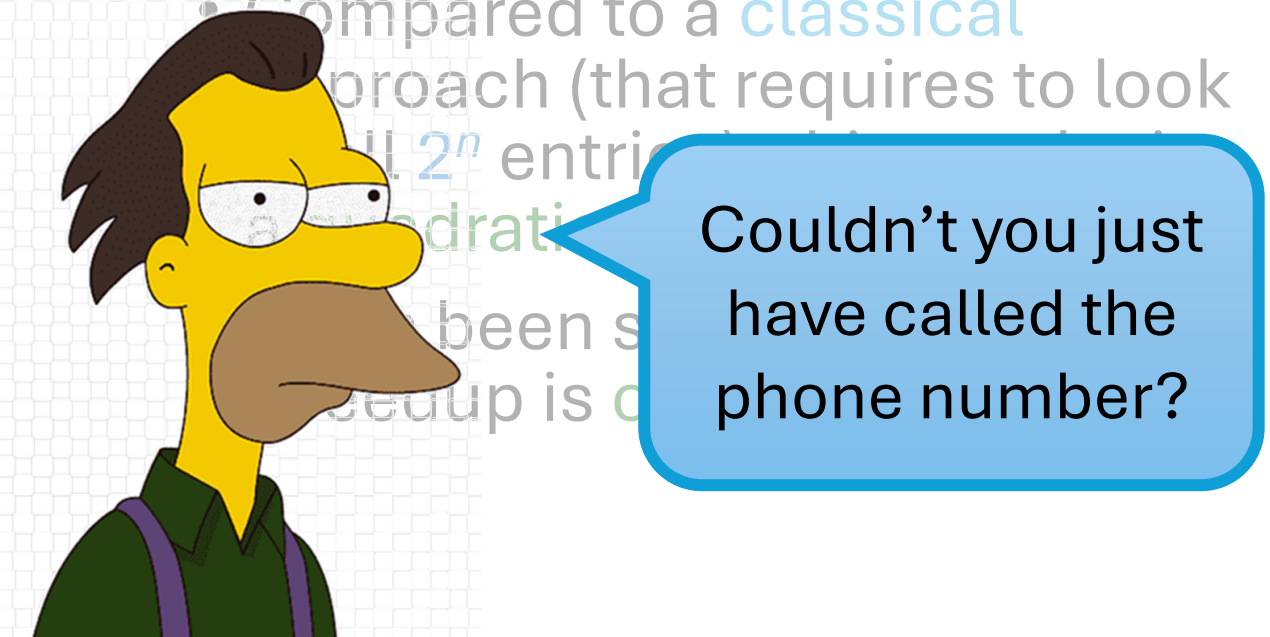
# Lov Grover to the rescue!

- In 1996, Grover published an algorithm that can find a target entry in an unstructured database that has $2^n$ entries by looking at only $\sqrt{2^n}$ entries.
- Compared to a classical approach (that requires to look at all $2^n$ entries), this results in a quadratic speedup!
- It has been shown that this speedup is optimal.



# Lov Grover to the rescue!

- In 1996, Grover published an algorithm that can find a target entry in an unstructured database that has $2^n$ entries by looking at only $\sqrt{2^n}$ entries.
- Compared to a classical approach (that requires to look all $2^n$ entries), Grover's algorithm has been shown to be optimal and its speedup is quadratic.



# Lov Grover to the rescue!

- In 1996, algorith… entry in … databas… looking a…
- …ompared to a classical …proach (t… …l $2^n$ e… …drati…
- …been …eedup is …

The key take-away here is that we may not get an exponential speedup (as was the case with Shor), but we can perhaps hope for a quadratic speedup when solving NP-Complete/Hard optimization problems.

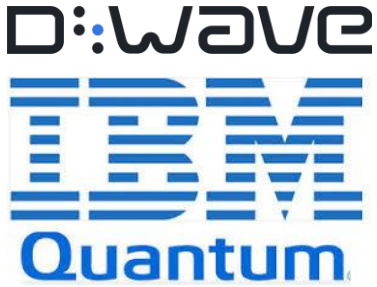
Ok, that's nice, however, even with a quadratic speedup, an NP-Complete problem is still NP Complete… Also, these results are almost 30 years old! Where do we stand now?

…over $\left(\sqrt{2^n}\right)$

quadratic sp…

Best clas…

# The NISQ era (Noisy Intermediate-Scale Quantum)




- Two big approaches for quantum computing:
  - Adiabatic quantum computing. Used by machines (quantum annealers) that have a single purpose: optimization! Example: D-Wave.
  - Gate-based universal quantum computing. Example: IBM. Note that a universal quantum computers is superior to a classical computer as any classical operation can be performed on a quantum computer with a polynomial overhead.
- Limitations shared by NISQ-era quantum computers:
  - Small number of qubits ➔ limited problem size.
  - Decoherence ➔ limited calculation time.
  - Noise ➔ limited accuracy/precision.

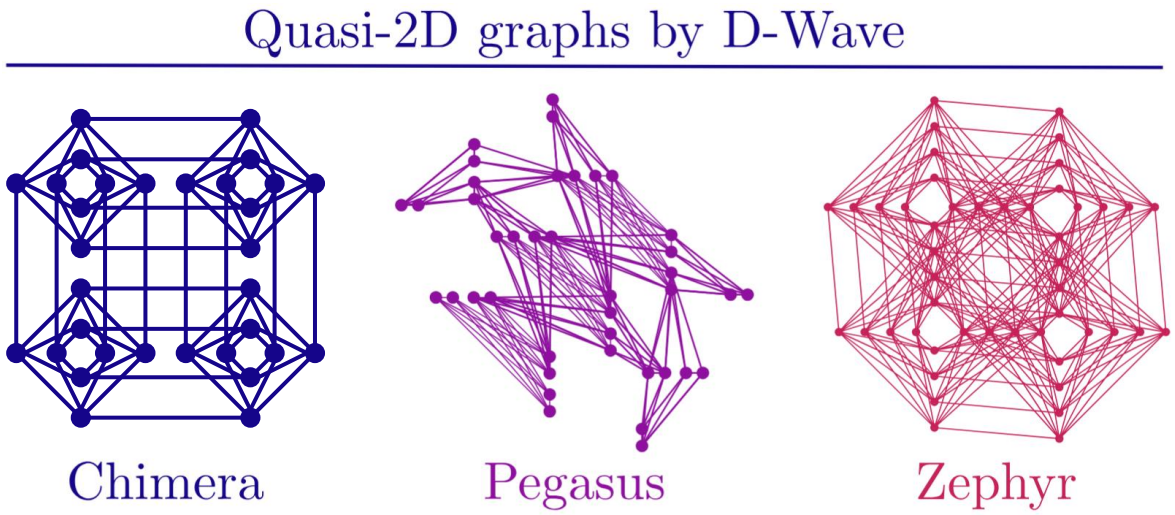
# The NISQ era (Noisy Intermediate-Scale Quantum)



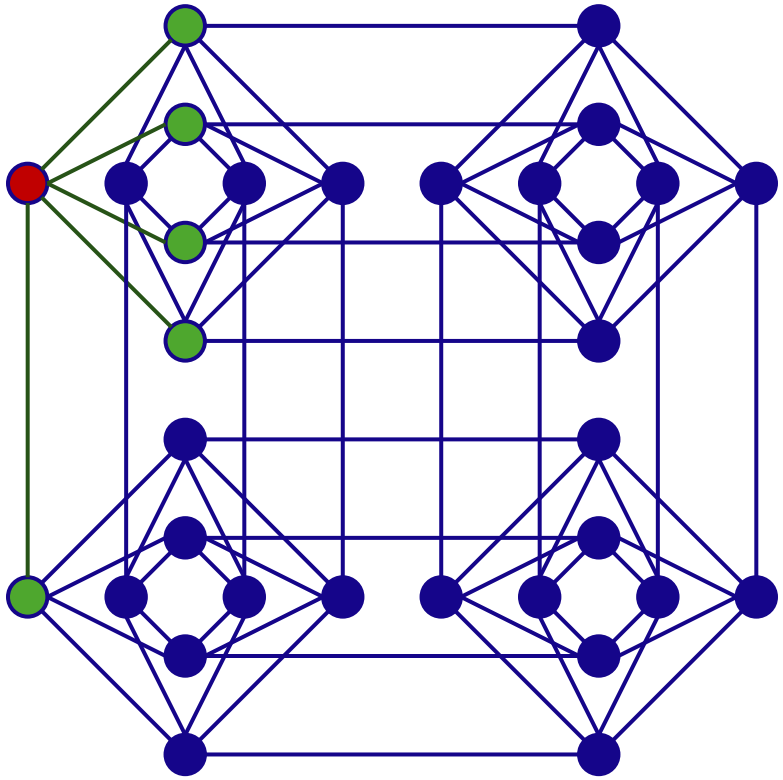

- Two big approaches for quantum computing:
  - Adiabatic quantum computing. Used by machines (quantum annealers) that have a single purpose: optimization! Example: D-Wave.
  - Gate-based universal quantum computing. Example: IBM. Note that a universal quantum computers is superior to a classical computer as any classical operation can be performed on a quantum computer with a polynomial overhead.
- Limitations shared by NISQ-era quantum computers:
  - Small number of qubits ➔ limited problem size.
  - Decoherence ➔ limited calculation time.
  - Noise ➔ limited accuracy/precision.

# Adiabatic quantum computing

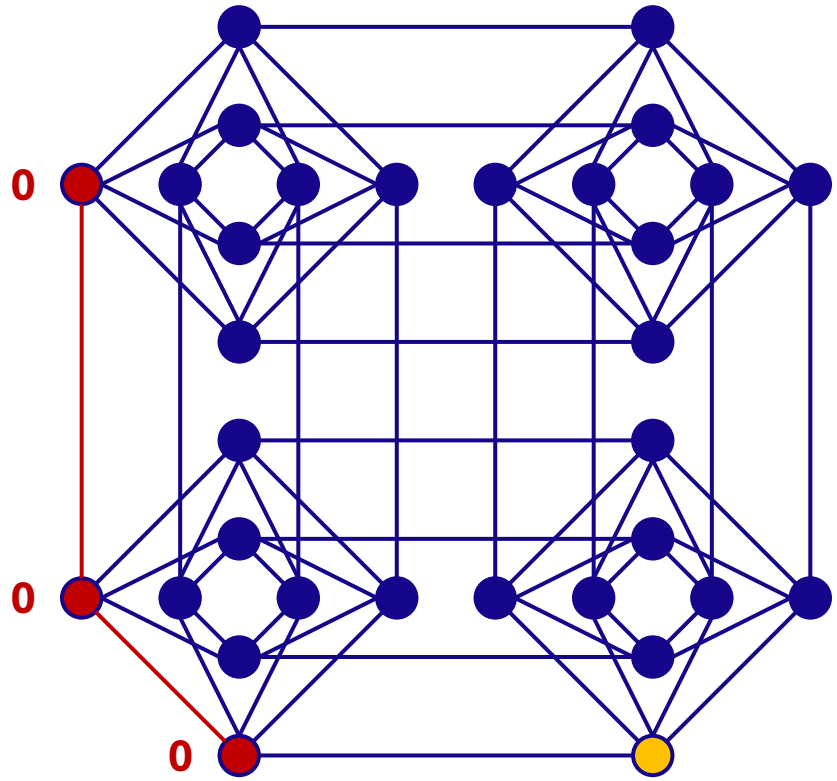
- Limitations of current quantum annealers:
  - Original optimization problem needs to be transformed to a problem that is understood by the quantum annealer (e.g., a QUBO).
  - QPU topology requires embedding.
  - Limited qubit connectivity.
  - Chains are required to connect qubits.



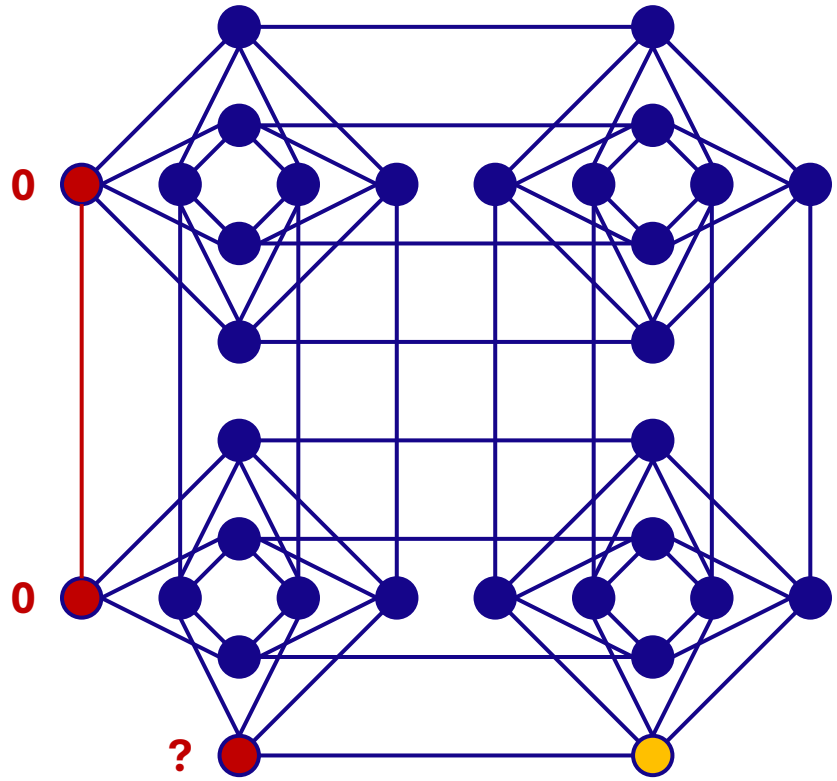
# Adiabatic quantum computing



# Adiabatic quantum computing



# Adiabatic quantum computing



# Adiabatic quantum computing

- Limitations of current quantum annealers:
  - Original optimization problem needs to be transformed to a problem that is understood by the quantum annealer (e.g., a QUBO).
  - QPU topology requires embedding.
  - Limited qubit connectivity.
  - Chains are required to connect qubits.
- Selected research:
  - Pérez Armas, Creemers, & Deleplanque. (2024). *Solving the resource constrained project scheduling problem with quantum annealing*, Nature Scientific Reports.
  - Deleplanque, Creemers, & Pérez Armas. (2025). *solQHealer: quantum procedures for rendering infeasible solutions feasible,* Journal of Heuristics.
  - Pérez Armas, Deleplanque, Aggoune, & Creemers (2025). *A quantum hybrid column-generation heuristic,* Philosophical Transactions A.
  - Saidi, Pérez Armas, Deleplanque, Hibti, & Creemers (2025). Exploring Quantum Annealing for Minimal Cut Sets Identification in Probabilistic Safety Assessment.

# Adiabatic quantum computing

- Limitations of current quantum annealers:
  - Original optimization problem needs to be transformed to a problem that is understood by the quantum annealer (e.g., a QUBO).
  - QPU topology requires embedding.
  - Limited qubit connectivity.
  - Chains are required to connect qubits.
- Selected research:
  - Pérez Armas, Creemers, & Deleplanque. (2024). *Solving the resource constrained project scheduling problem with quantum annealing*, Nature Scientific Reports.
  - Deleplanque, Creemers, & Pérez Armas. (2025). *solQHealer: quantum procedures for rendering infeasible solutions feasible,* Journal of Heuristics.
  - Pérez Armas, Deleplanque, Aggoune, & Creemers (2025). *A quantum hybrid column-generation heuristic,* Philosophical Transactions A.
  - Saidi, Pérez Armas, Deleplanque, Hibti, & Creemers (2025). Exploring Quantum Annealing for Minimal Cut Sets Identification in Probabilistic Safety Assessment.

# Adiabatic quantum computing

(selected research 1)

- Pérez Armas, Creemers, & Deleplanque. (2024). *Solving the Resource Constrained Project Scheduling Problem (RCPSP) with quantum annealing*.
- What: we use quantum annealing to compare 12 well-known classical formulations for solving the RCPSP.
- Key take-away: formulations that work well on classical computers do not necessarily work well on quantum computers. In fact, on a quantum computer, the oldest formulation (which required the least number of qubits) had the best performance.

# Adiabatic quantum computing

- Limitations of current quantum annealers:
  - Original optimization problem needs to be transformed to a problem that is understood by the quantum annealer (e.g., a QUBO).
  - QPU topology requires embedding.
  - Limited qubit connectivity.
  - Chains are required to connect qubits.
- Selected research:
  - Pérez Armas, Creemers, & Deleplanque. (2024). *Solving the resource constrained project scheduling problem with quantum annealing*, Nature Scientific Reports.
  - Deleplanque, Creemers, & Pérez Armas. (2025). *solQHealer: quantum procedures for rendering infeasible solutions feasible,* Journal of Heuristics.
  - Pérez Armas, Deleplanque, Aggoune, & Creemers (2025). *A quantum hybrid column-generation heuristic,* Philosophical Transactions A.
  - Saidi, Pérez Armas, Deleplanque, Hibti, & Creemers (2025). Exploring Quantum Annealing for Minimal Cut Sets Identification in Probabilistic Safety Assessment.

# Adiabatic quantum computing

(selected research 2)

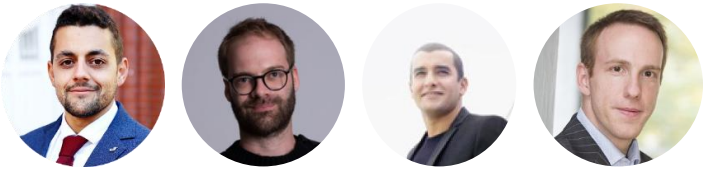
- Deleplanque, Creemers, & Pérez Armas. (2025). *solQHealer: quantum procedures for rendering infeasible solutions feasible*.
- What: we use (reverse) quantum annealing to solve the Maximum Independent Set (MIS) problem as well as 3-SAT.
- Key take-away: (reverse) quantum annealing may be used to quickly repair infeasible solutions/solutions that have become infeasible due to new constraints that have surfaced. This is particularly useful in a setting where fast, online optimization is required (e.g., train scheduling or portfolio optimization).

# Adiabatic quantum computing

- Limitations of current quantum annealers:
  - Original optimization problem needs to be transformed to a problem that is understood by the quantum annealer (e.g., a QUBO).
  - QPU topology requires embedding.
  - Limited qubit connectivity.
  - Chains are required to connect qubits.
- Selected research:
  - Pérez Armas, Creemers, & Deleplanque. (2024). *Solving the resource constrained project scheduling problem with quantum annealing*, Nature Scientific Reports.
  - Deleplanque, Creemers, & Pérez Armas. (2025). *solQHealer: quantum procedures for rendering infeasible solutions feasible,* Journal of Heuristics.
  - Pérez Armas, Deleplanque, Aggoune, & Creemers (2025). *A quantum hybrid column-generation heuristic,* Philosophical Transactions A.
  - Saidi, Pérez Armas, Deleplanque, Hibti, & Creemers (2025). Exploring Quantum Annealing for Minimal Cut Sets Identification in Probabilistic Safety Assessment.

# Adiabatic quantum computing

(selected research 3)



- Pérez Armas, Deleplanque, Aggoune, & Creemers (2025). *A quantum hybrid column-generation procedure*.
- What: we use a hybrid column-generation procedure to solve the parallel machine scheduling problem as well as the 2-dimensional cutting stock problem.
- Key take-away: quantum annealers excel in rapidly generating many (good) solutions. These solutions may, for instance, be introduced as new columns in a hybrid column generation procedure (where the master problem is solved by a classical computer).

# Adiabatic quantum computing

- Limitations of current quantum annealers:
  - Original optimization problem needs to be transformed to a problem that is understood by the quantum annealer (e.g., a QUBO).
  - QPU topology requires embedding.
  - Limited qubit connectivity.
  - Chains are required to connect qubits.
- Selected research:
  - Pérez Armas, Creemers, & Deleplanque. (2024). *Solving the resource constrained project scheduling problem with quantum annealing*, Nature Scientific Reports.
  - Deleplanque, Creemers, & Pérez Armas. (2025). *solQHealer: quantum procedures for rendering infeasible solutions feasible,* Journal of Heuristics.
  - Pérez Armas, Deleplanque, Aggoune, & Creemers (2025). *A quantum hybrid column-generation heuristic,* Philosophical Transactions A.
  - Saidi, Pérez Armas, Deleplanque, Hibti, & Creemers (2025). Exploring Quantum Annealing for Minimal Cut Sets Identification in Probabilistic Safety Assessment.

# Adiabatic quantum computing

(selected research 4)

- Saidi, Pérez Armas, Deleplanque, Hibti, & Creemers (2025). *Exploring Quantum Annealing for Minimal Cut Sets Identification in Probabilistic Safety Assessment*.
- What: we compare different D-Wave topologies (Pegasus vs Zephyr) and different embedding methods to verify whether quantum annealing can be used to solve practical use-cases at Électricité De France (EDF).
- Key take-aways:
  - Results depend heavily on embedding.
  - In general, the new topology (Zephyr) is better than the old topology (Pegasus).
  - Classical method (simulated annealing) often provides better solutions, however, quantum annealing is incredibly fast (more than 100 times faster)!

# The NISQ era (Noisy Intermediate-Scale Quantum)

- Two big approaches for quantum computing:
  - Adiabatic quantum computing. Used by machines (quantum annealers) that have a single purpose: optimization! Example: D-Wave.
  - Gate-based universal quantum computing. Example: IBM. Note that a al computer as any omputer with a
- ers:

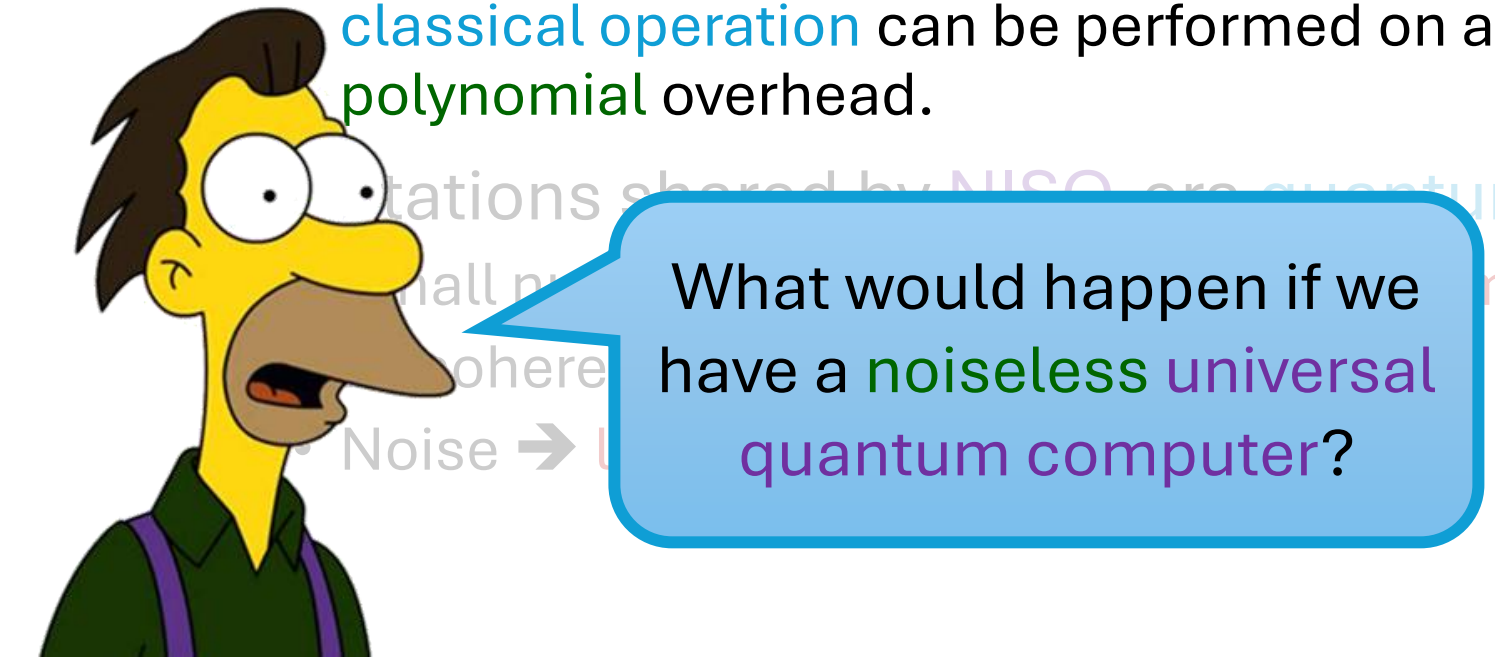
Lessons learned:

1. Don't expect that what works in the classical world also works in a quantum world.
2. Current machines may not yet be advanced enough to find optimal solutions for big, real-life problems. However, they can already be used to obtain fast, heuristic solutions or in hybrid procedures that need fast (but good) solutions.

# The NISQ era (Noisy Intermediate-Scale Quantum)




- Two big approaches for quantum computing:
  - Adiabatic quantum computing. Used by machines (quantum annealers) that have a single purpose: optimization! Example: D-Wave.
  - Gate-based universal quantum computing. Example: IBM. Note that a universal quantum computers is superior to a classical computer as any classical operation can be performed on a quantum computer with a polynomial overhead.




- tations shared by NISQ-era quantum computers:
  - hall n… …n size.
  - ohere…
  - Noise ➔ l…

# Universal quantum computing

- A noiseless universal quantum computer does not exist (yet)…
- However, we can simulate one!
- We coded an ideal simulator to assess the performance of future quantum optimization algorithms.
- Selected research:
  - Creemers & Pérez Armas. (2025). *Discrete optimization: a quantum revolution?, EJOR*.
  - Creemers & Pérez Armas. (2025). Limitations of existing quantum algorithms, Philosophical Transactions A.
  - Creemers. (2025). *Speeding up Grover's algorithm, EJOR*.
  - Creemers. (2025). *Optimal performance when solving discrete optimization problems?*

# Universal quantum computing

- A noiseless universal quantum computer does not exist (yet)…
- However, we can simulate one!
- We coded an ideal simulator to assess the performance of future quantum optimization algorithms.
- Selected research:
  - Creemers & Pérez Armas. (2025). *Discrete optimization: a quantum revolution?, EJOR*.
  - Creemers & Pérez Armas. (2025). Limitations of existing quantum algorithms, Philosophical Transactions A.
  - Creemers. (2025). *Speeding up Grover's algorithm, EJOR*.
  - Creemers. (2025). *Optimal performance when solving discrete optimization problems?*

# Universal quantum computing

(selected research 1)

- Creemers & Pérez Armas. (2025). *Discrete optimization: a quantum revolution?*
- What: we develop several Grover-based quantum procedures for solving discrete optimization problems.
- Key take-aways:
  - We can solve any discrete optimization problem using $O\left(\mu\sqrt{2^{bn}}\right)$ operations, where $\mu$ is the number of operations required to verify a solution & $2^b$ is the number of discrete values that can be assigned to any of the $n$ decision variables.
  - Our procedures can be used as general-purpose solvers (similar to CPLEX and Gurobi) but also as heuristics.
  - We present a hybrid Branch-and-Bound (B&B) procedure that expects to visit $O\left(\sqrt{2^n}\right)$ nodes. In contrast, in the worst case, a classical B&B visits $O(2^n)$ nodes ➔ we achieve a quadratic speedup!
  - We demonstrate that our procedures can match the worst-case performance of the best classical algorithms that solve the knapsack problem.
  - For quadratic knapsack problems we outperform the best classical algorithms.

# Quadratic knapsack

$$\max \sum_{i=1}^{n} \sum_{j=1}^{n} v_{ij} x_i x_j$$

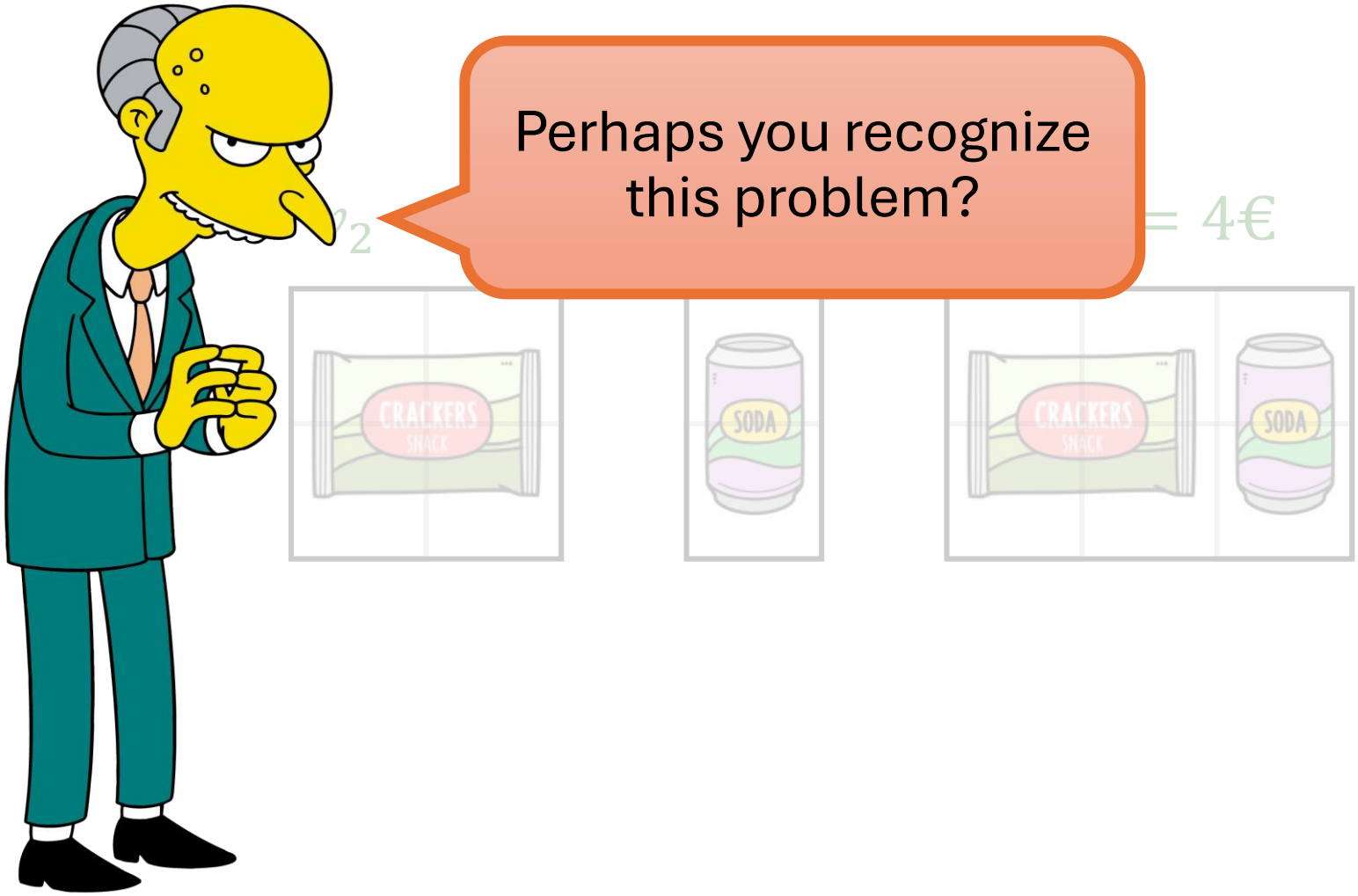
$$subject\ to: \sum_{i=1}^{n} w_i x_i \leq W$$

$v_2 = 6€$ $v_6 = 2€$ $v_{26} = 4€$

# Quadratic knapsack

$$\max \sum_{i=1}^{n} \sum_{j=1}^{n} v_{ij} x_i x_j$$

$$subject\ to: \sum_{i=1}^{n} w_i x_i \leq W$$



# Quadratic knapsack

$$\max \sum_{i=1}^{n} \sum_{j=1}^{n} v_{ij} x_i x_j$$

$$subject\ to: \sum_{i=1}^{n} w_i x_i \leq W$$

$$\sum_{i=1}^{n} v_i x_i \geq V$$

# Quadratic knapsack = Markowitz model

$$\max \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} x_i x_j$$

$$subject\ to: \sum_{i=1}^{n} w_i x_i \leq W$$

$$\sum_{i=1}^{n} v_i x_i \geq V$$

Where $\sigma_{ij}$ is the covariance between the return on asset of item (i.e., asset) $i$ and item (i.e., asset) $j$.

- Foundation of Modern Portfolio Theory (MPT).
- Winner of Nobel prize in economics!

# Universal quantum computing

- A noiseless universal quantum computer does not exist (yet)…
- However, we can simulate one!
- We coded an ideal simulator to assess the performance of future quantum optimization algorithms.
- Selected research:
  - Creemers & Pérez Armas. (2025). *Discrete optimization: a quantum revolution?, EJOR.*
  - Creemers & Pérez Armas. (2025). Limitations of existing quantum algorithms, Philosophical Transactions A.
  - Creemers. (2025). *Speeding up Grover's algorithm, EJOR.*
  - Creemers. (2025). *Optimal performance when solving discrete optimization problems?*

# Universal quantum computing

(selected research 2)

- Creemers & Pérez Armas. (2025). Limitations of existing quantum algorithms.
- What: we investigate whether we can use quantum counting, nested quantum search, and amplitude amplification for solving optimization problems.
- Key take-aways:
  - When effectively implementing these quantum algorithms, several challenges need to be overcome.
  - They do not allow to outperform Vanilla Grover; they do not allow a better-than quadratic speedup.

# Universal quantum computing

- A noiseless universal quantum computer does not exist (yet)…
- However, we can simulate one!
- We coded an ideal simulator to assess the performance of future quantum optimization algorithms.
- Selected research:
  - Creemers & Pérez Armas. (2025). *Discrete optimization: a quantum revolution?, EJOR.*
  - Creemers & Pérez Armas. (2025). Limitations of existing quantum algorithms, Philosophical Transactions A.
  - Creemers. (2025). *Speeding up Grover's algorithm, EJOR.*
  - Creemers. (2025). *Optimal performance when solving discrete optimization problems?*

# Universal quantum computing

(selected research 3)

- Creemers. (2025). *Speeding up Grover's algorithm*.
- What: we assess the (expected) speedup when running Grover's algorithm in series, parallel, and in series/parallel.
- Key take-aways:
  - The expected runtime can be reduced by almost 13% if we run Grover's algorithm in series.
  - The parallel execution of Grover's algorithm yields only a square-root speedup in the number of QPUs (e.g., if we use $4$ QPUs, we only obtain a speedup of factor $\sqrt{4} = 2$).
  - Partitioning the search space (i.e., the database) is not a dominant strategy to speedup Grover's algorithm.

# Universal quantum computing

- A noiseless universal quantum computer does not exist (yet)…
- However, we can simulate one!
- We coded an ideal simulator to assess the performance of future quantum optimization algorithms.
- Selected research:
  - Creemers & Pérez Armas. (2025). *Discrete optimization: a quantum revolution?, EJOR.*
  - Creemers & Pérez Armas. (2025). Limitations of existing quantum algorithms, Philosophical Transactions A.
  - Creemers. (2025). *Speeding up Grover's algorithm, EJOR.*
  - Creemers. (2025). *Optimal performance when solving discrete optimization problems?*

# Universal quantum computing

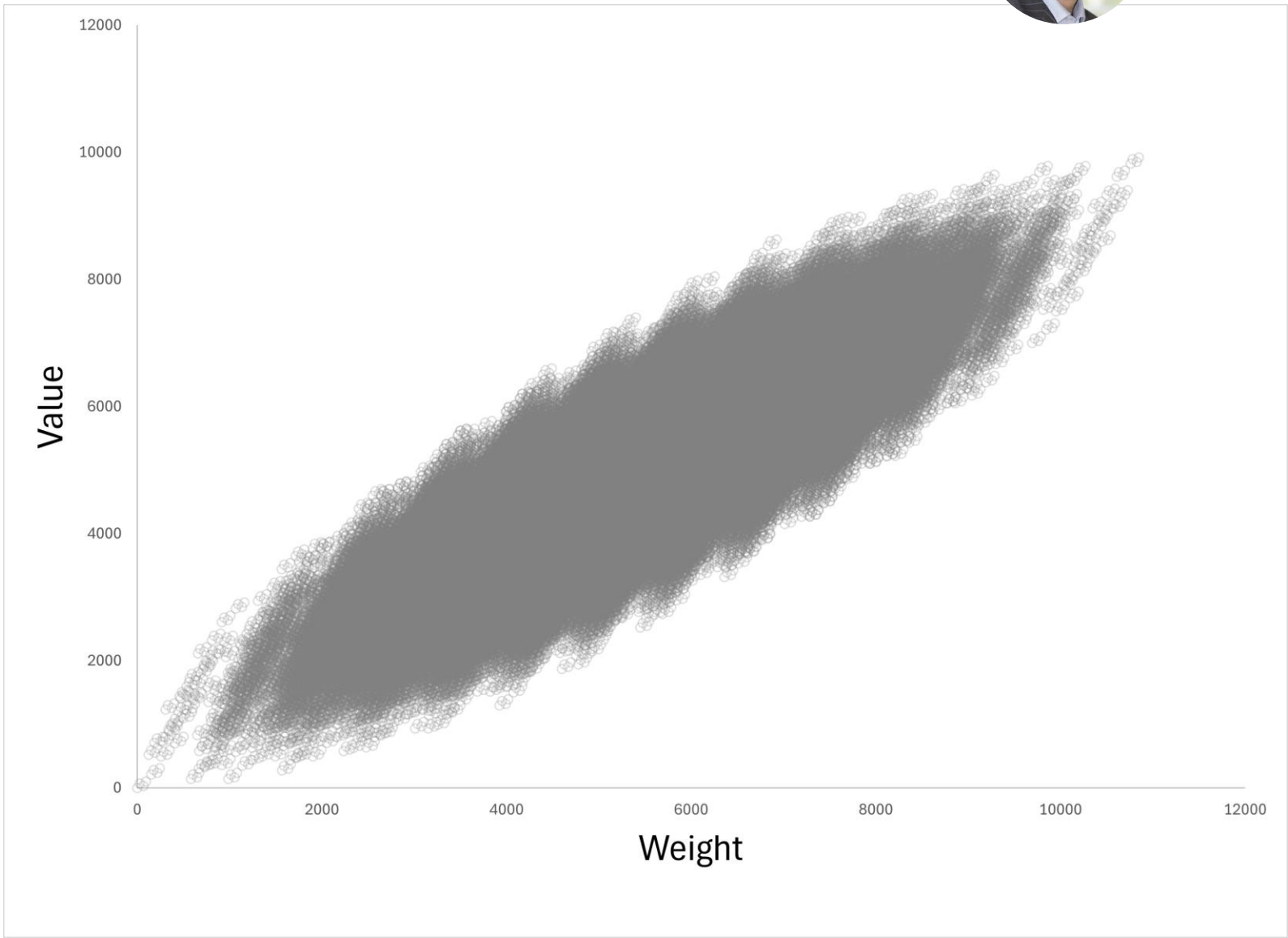
(selected research 4)

- Creemers. (2025). *Optimal performance when solving discrete optimization problems?*
- What: if $P$ denotes the class of all discrete optimization problems, we consider a class of problems $P' \subseteq P$. Note that:
  - $P'$ contains many well-known problems such as knapsack, TSP, RCPSP…
  - Any problem in $P$ can be transformed into a problem in $P'$.
- For problems in $P'$:
  - We obtain a superposition of locally optimal solutions in polynomial time.
  - We can use this superposition to find a globally optimal solution.

# Universal quantum computing

(selected research 4)

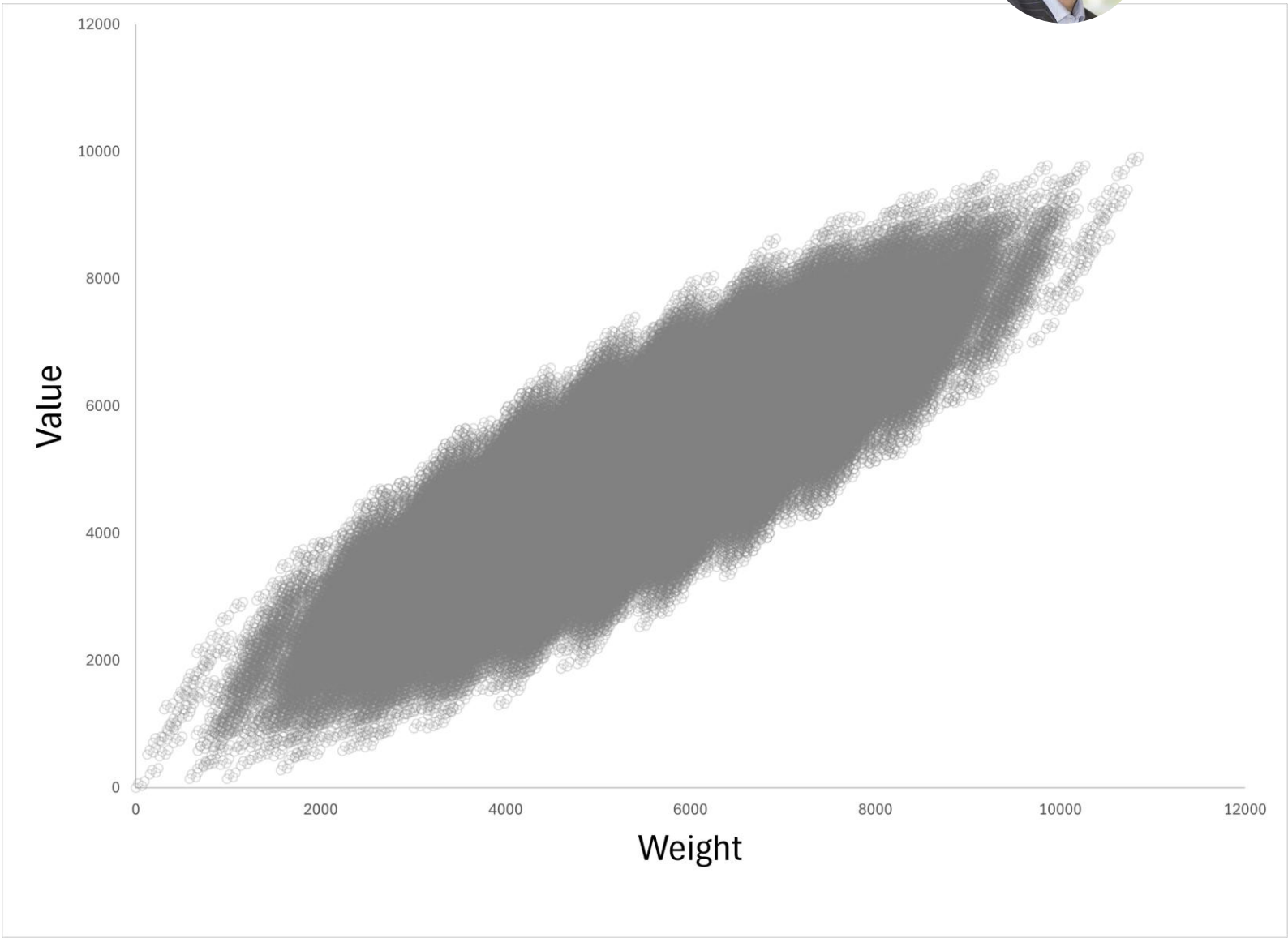
- Example knapsack problem that has $n = 19$ items (i.e., $2^{19} = 524288$ solutions).



# Universal quantum computing

(selected research 4)

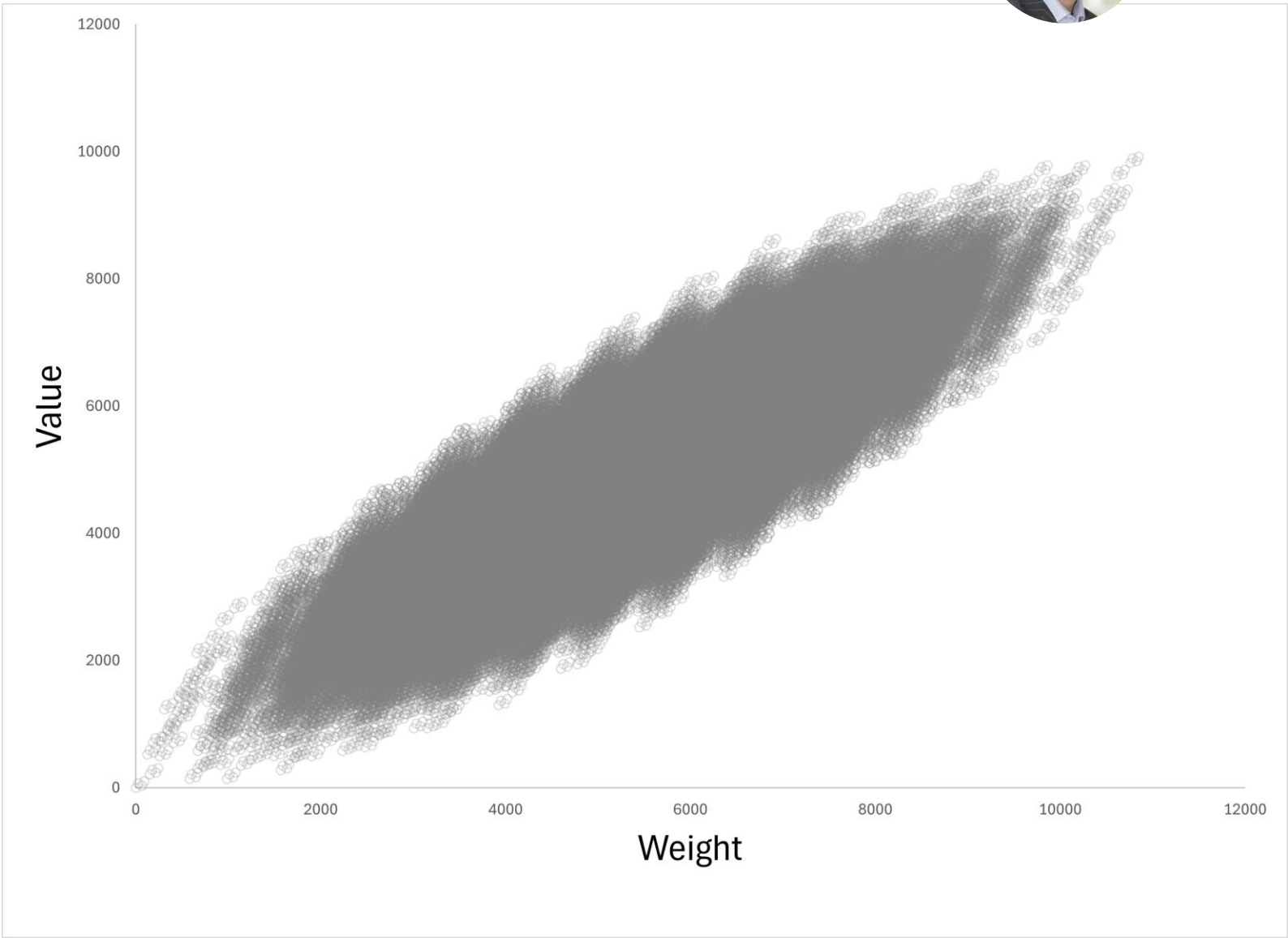
- Example knapsack problem that has $n = 19$ items (i.e., $2^{19} = 524288$ solutions).
- Each dot represents a solution that has a value and a weight.
- We start from a uniform superposition where each solution has the same probability (i.e., $1/524288$) of being measured.



# Universal quantum computing

(selected research 4)

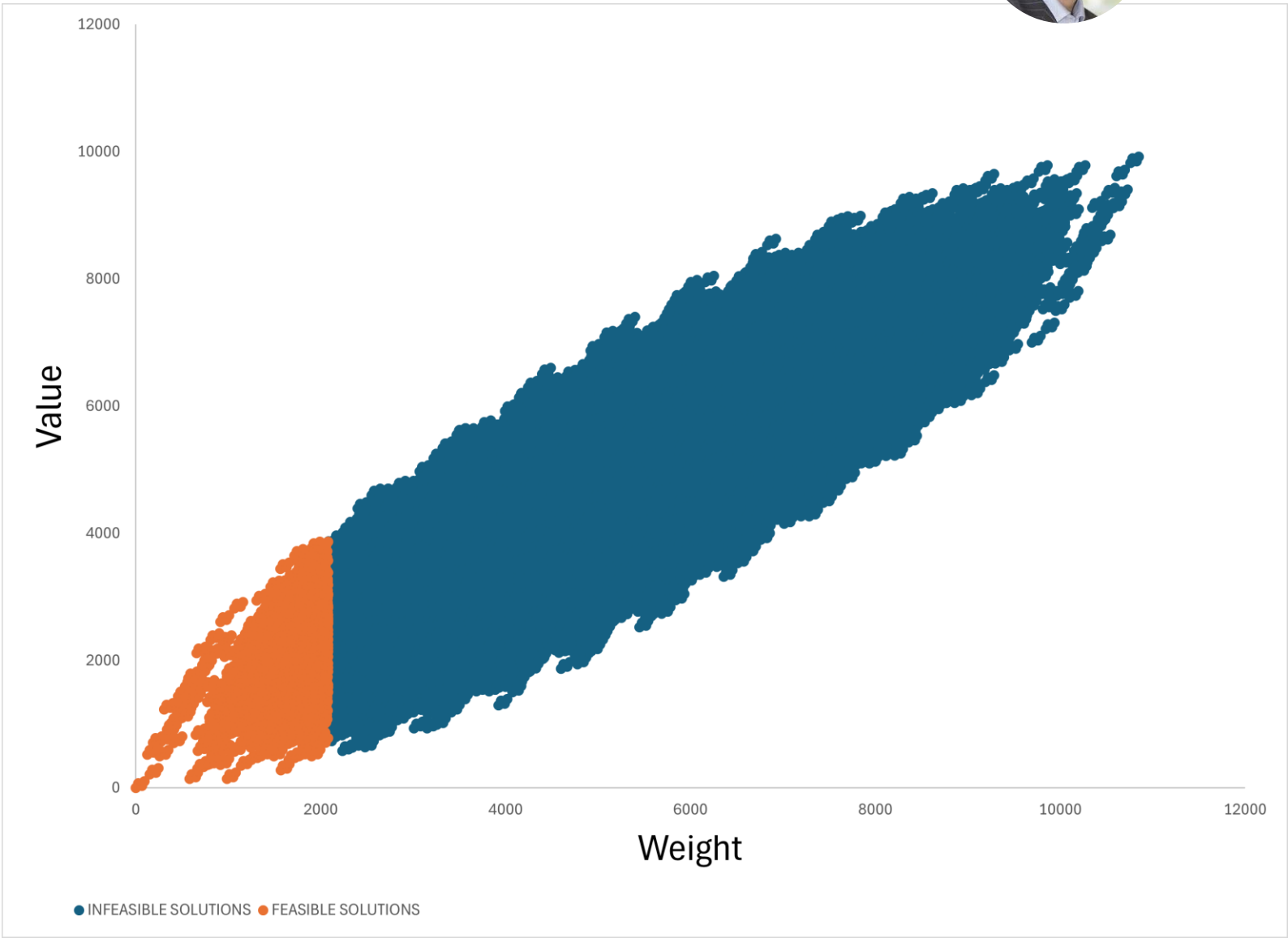
- The weight capacity of the knapsack is $W = 2081$.



# Universal quantum computing

(selected research 4)

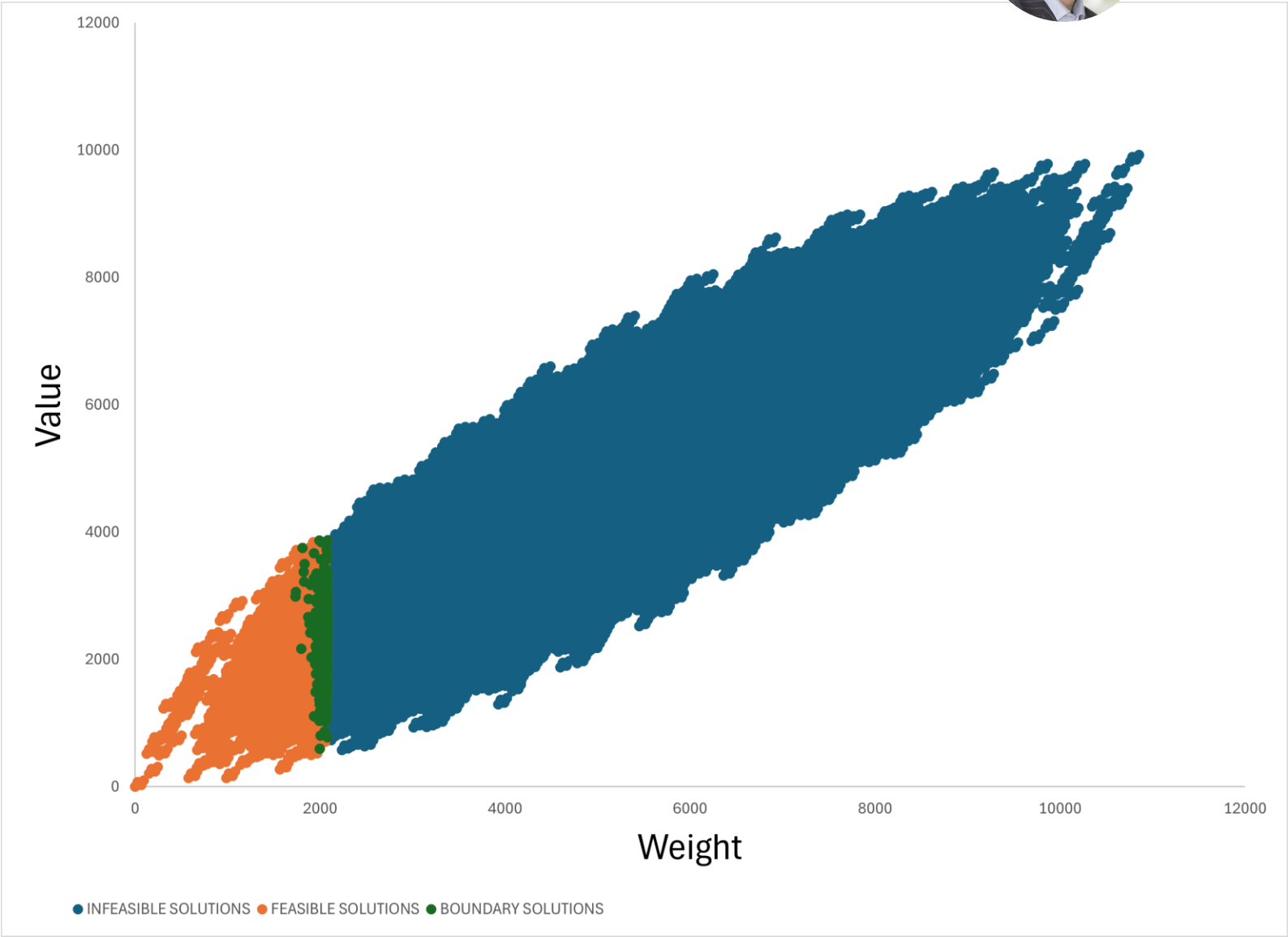
- The weight capacity of the knapsack is $W = 2081$.
- Infeasible solutions have a weight larger than $W$.
- Feasible solutions have a weight smaller than $W$.
- Question: how do we obtain a superposition of all feasible solutions?



# Universal quantum computing

(selected research 4)

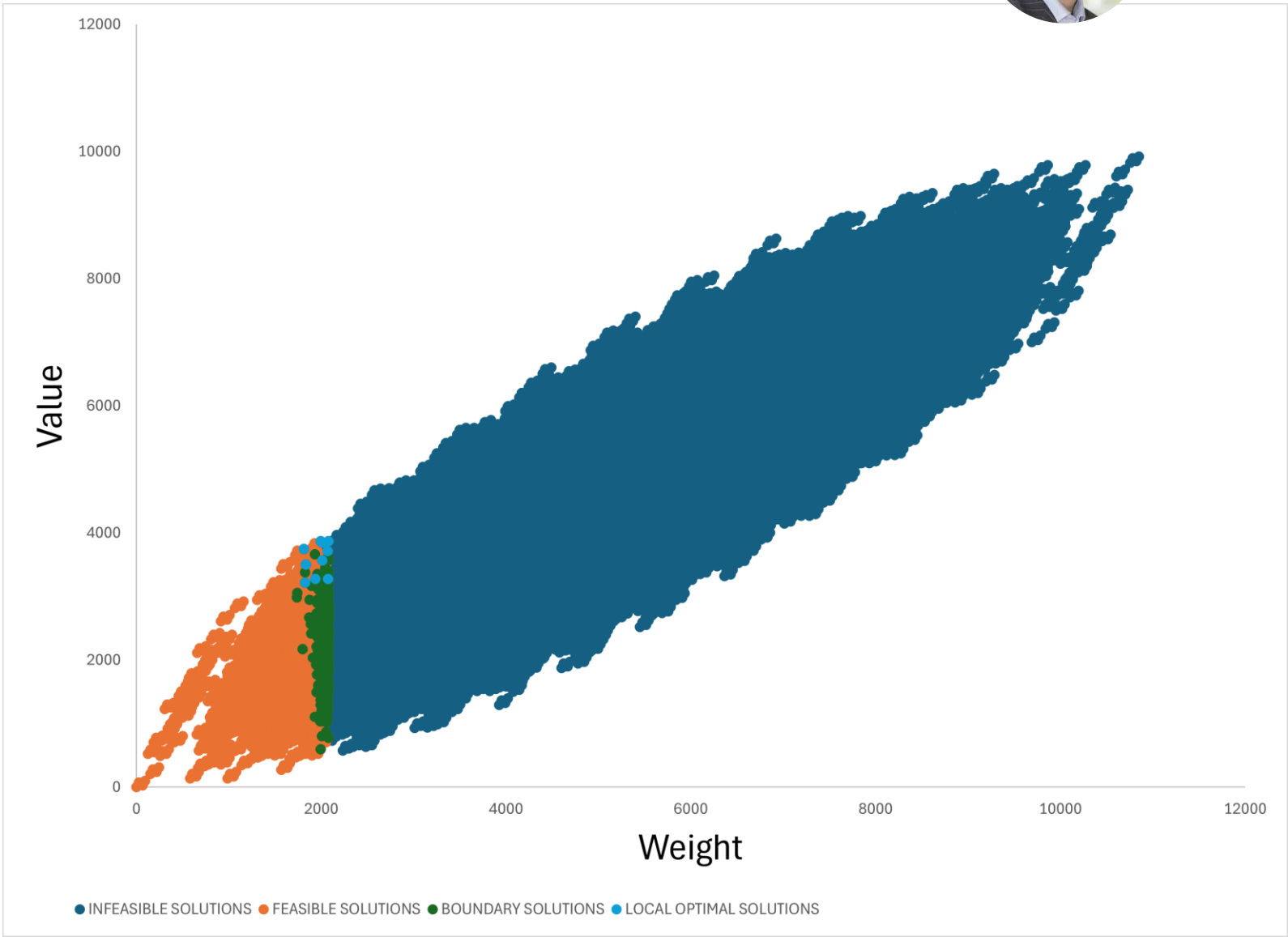
- In fact, we can do better! We can obtain a superposition of boundary solutions.
- Boundary solutions are feasible solutions for which no item can still be added without exceeding $W$.
- This superposition can be obtained in polynomial time!



# Universal quantum computing

(selected research 4)

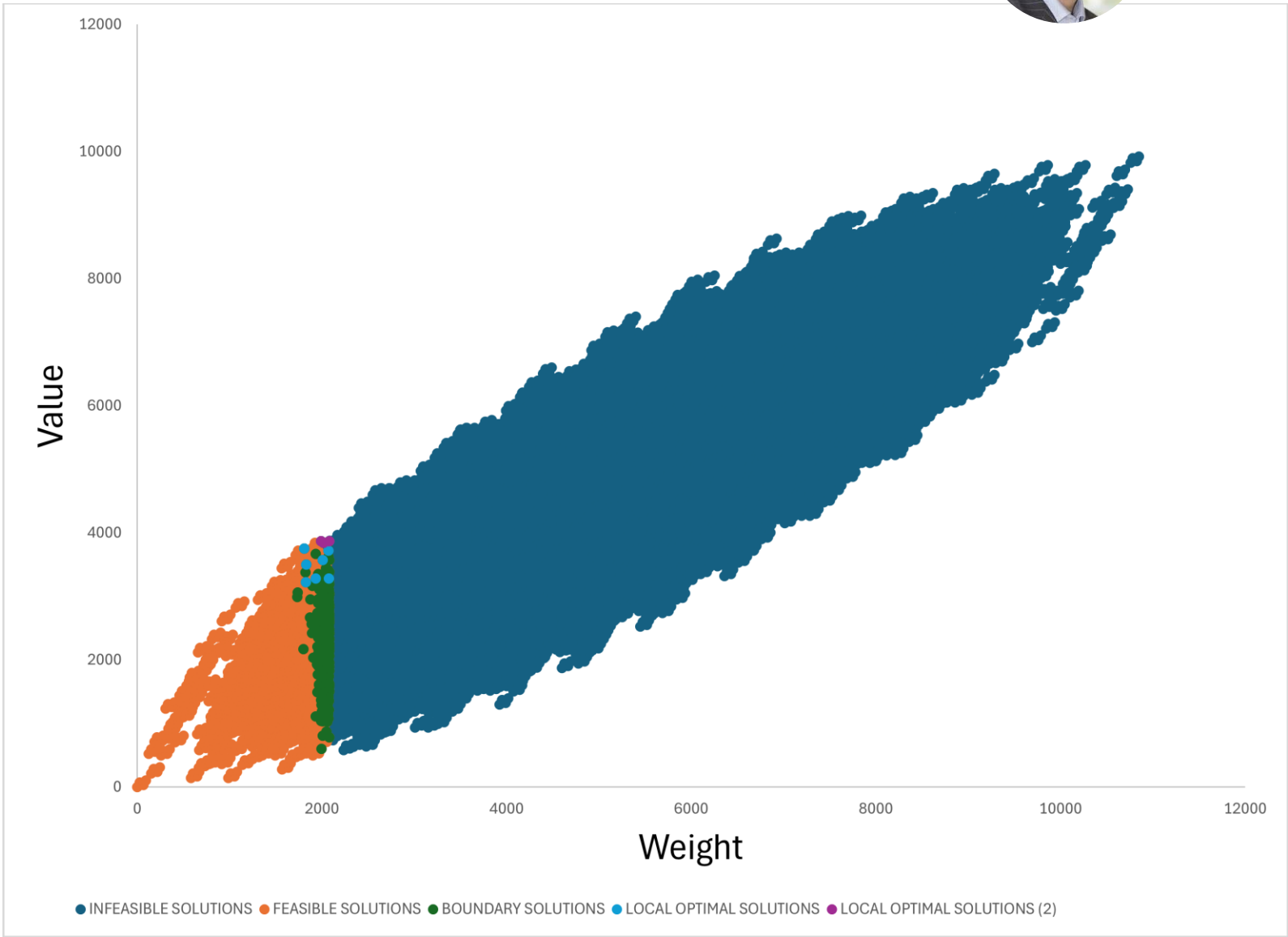
- We can take it one step further and can remove all “bad” boundary solutions such that only locally optimal solutions remain.
- Once again, we can obtain this superposition of locally optimal solutions in polynomial time!



# Universal quantum computing

(selected research 4)

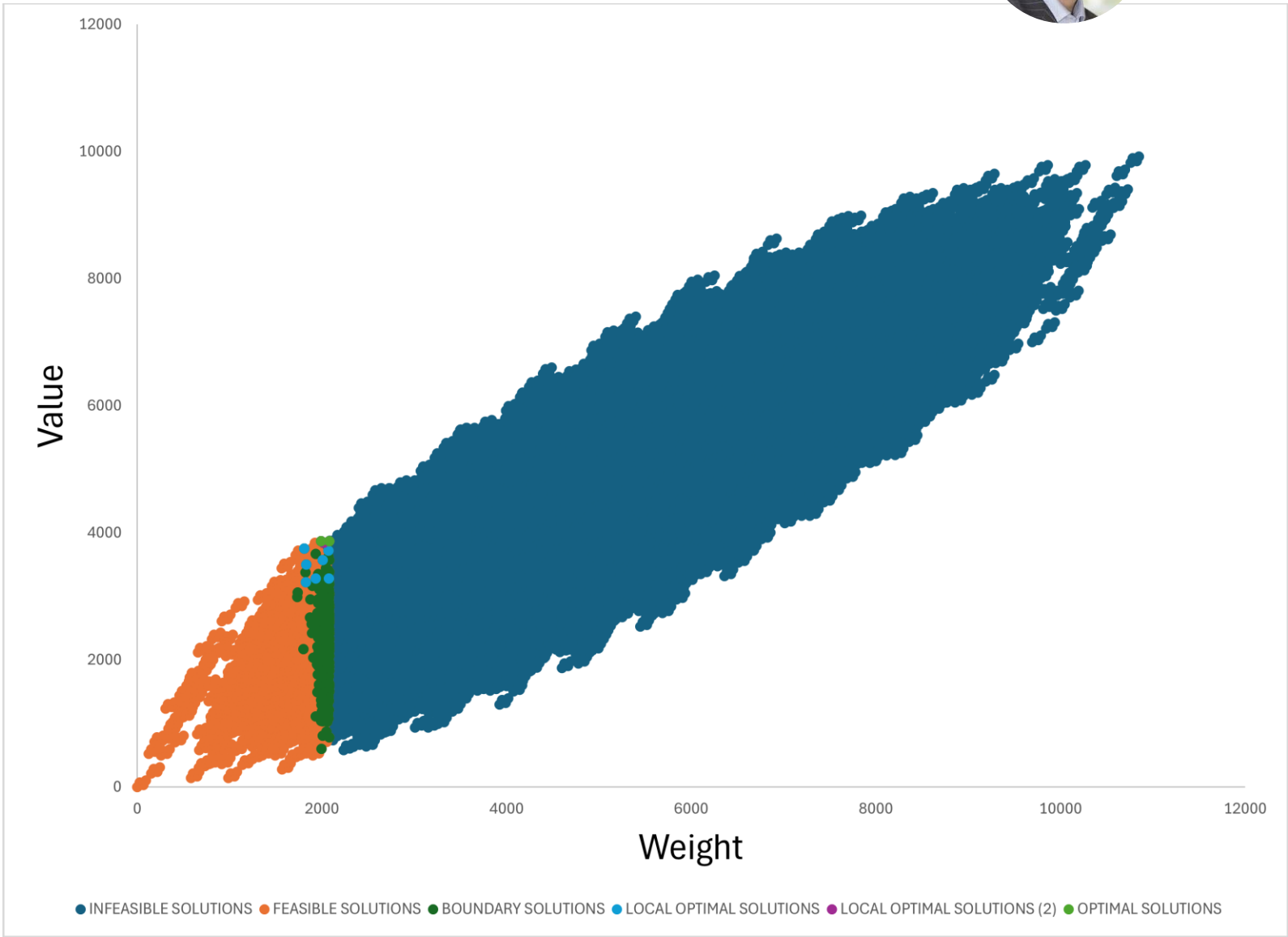
- If we use problem-specific dominance- and priority rules, we can do even better and can obtain a subset of locally optimal solutions.
- We can obtain this superposition in polynomial time!
- In this example, there are two globally optimal solutions (the subset contains three locally optimal solutions).



# Universal quantum computing

(selected research 4)

- If we use problem-specific dominance- and priority rules, we can do even better and can obtain a subset of locally optimal solutions.
- We can obtain this superposition in polynomial time!
- In this example, there are two globally optimal solutions (the subset contains three locally optimal solutions).



# Universal quantum computing

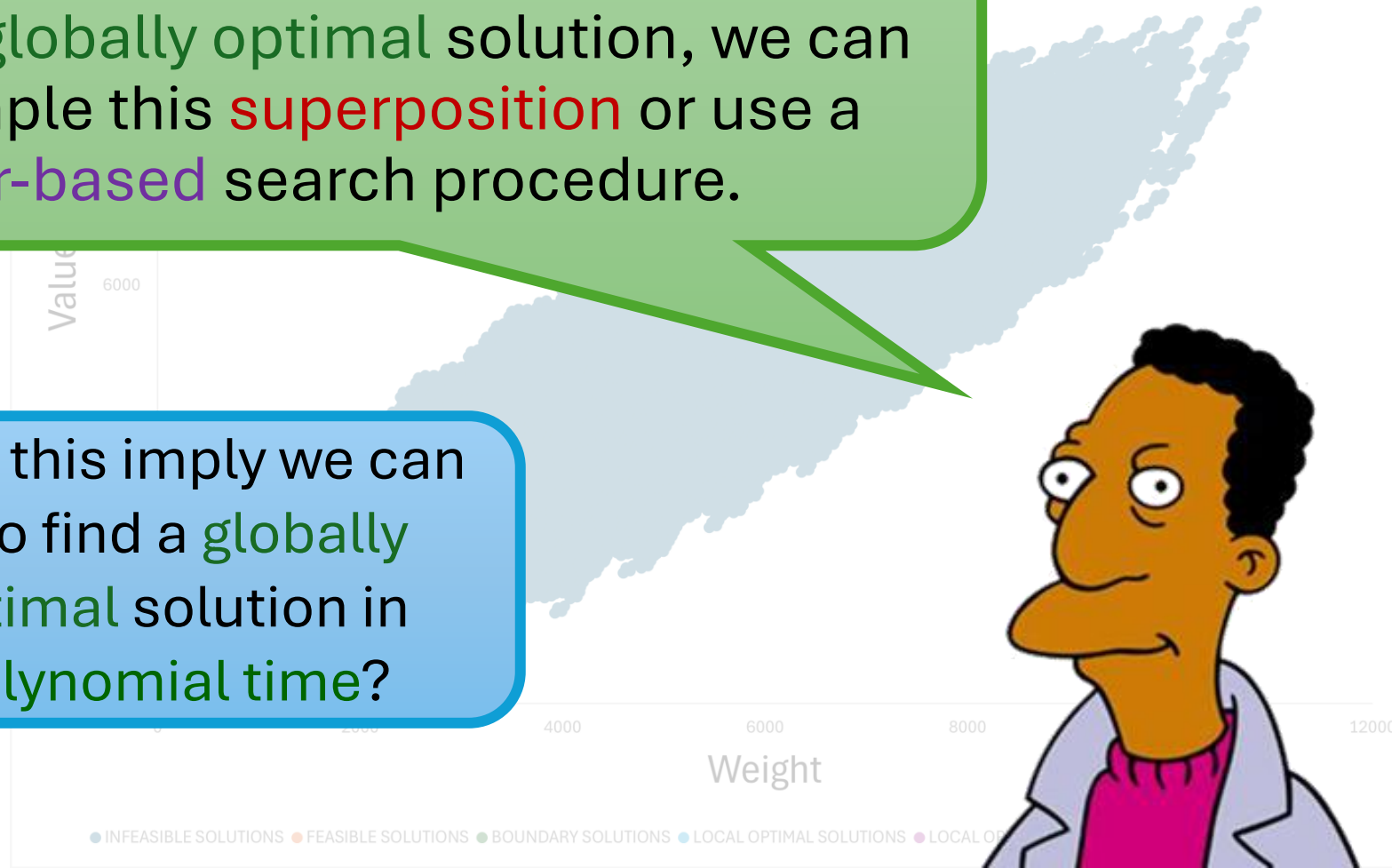
(selected research 4)



- If we use prob
  dominance- a
  rules, we can
  and can obta
  locally optimal
- W n obtain this
  sition in polynomial
- In
  g lly o soluti
  se tains t
  timal solutions).

Wow! We can obtain a superposition of "good" locally optimal solutions in polynomial time! To obtain a globally optimal solution, we can either sample this superposition or use a Grover-based search procedure.

Does this imply we can also find a globally optimal solution in polynomial time?



# Universal quantum computing

(select

- If we use p
  dominanc
  rules, we
  and can o
  locally opt
- W n obtain this
  sition in polynomial
- In
  g lly o soluti
  se tains t
  timal solutions).



Unfortunately, no… There can still be an exponential number of "good" locally optimal solutions. However, this set of "good" locally optimal solutions may be seen as an unstructured database (i.e., not much problem structure is left that can still be exploited to find the globally optimal solution).

Aha, and this is important why?

# Universal quantum computing

(selected research 4)

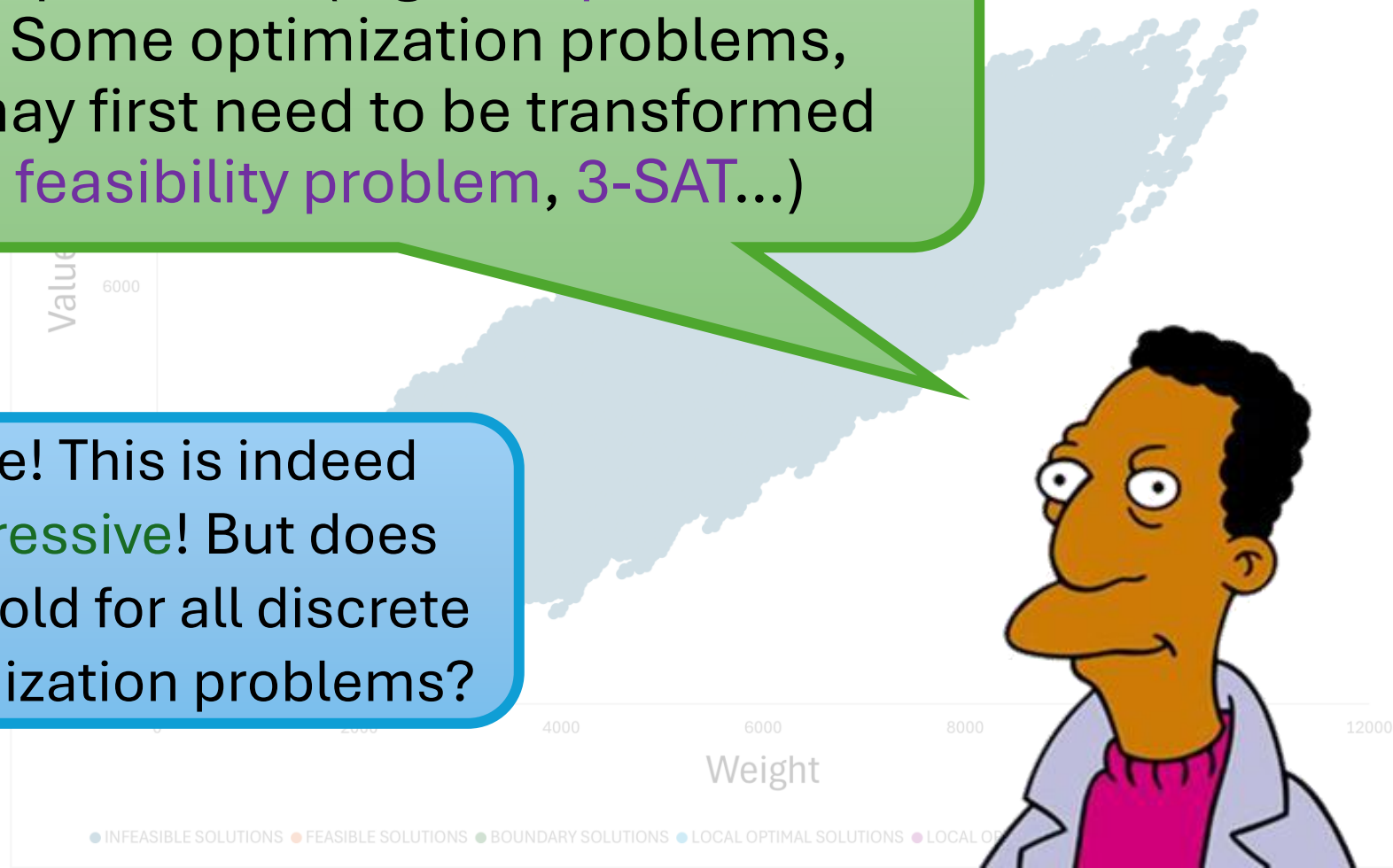


- If we use prob
  dominance- a
  rules, we can
  and can obta
  locally optima
- W n obtain this
  sition in polynomial
- In
  g lly o soluti
  se tains th
  timal solutions).

This result holds for many well-known optimization problems (e.g., knapsack, TSP, RCPSP…). Some optimization problems, however, may first need to be transformed (e.g., 0-1 feasibility problem, 3-SAT…)

I see! This is indeed impressive! But does this hold for all discrete optimization problems?





# Universal quantum computing

(selected research 4)

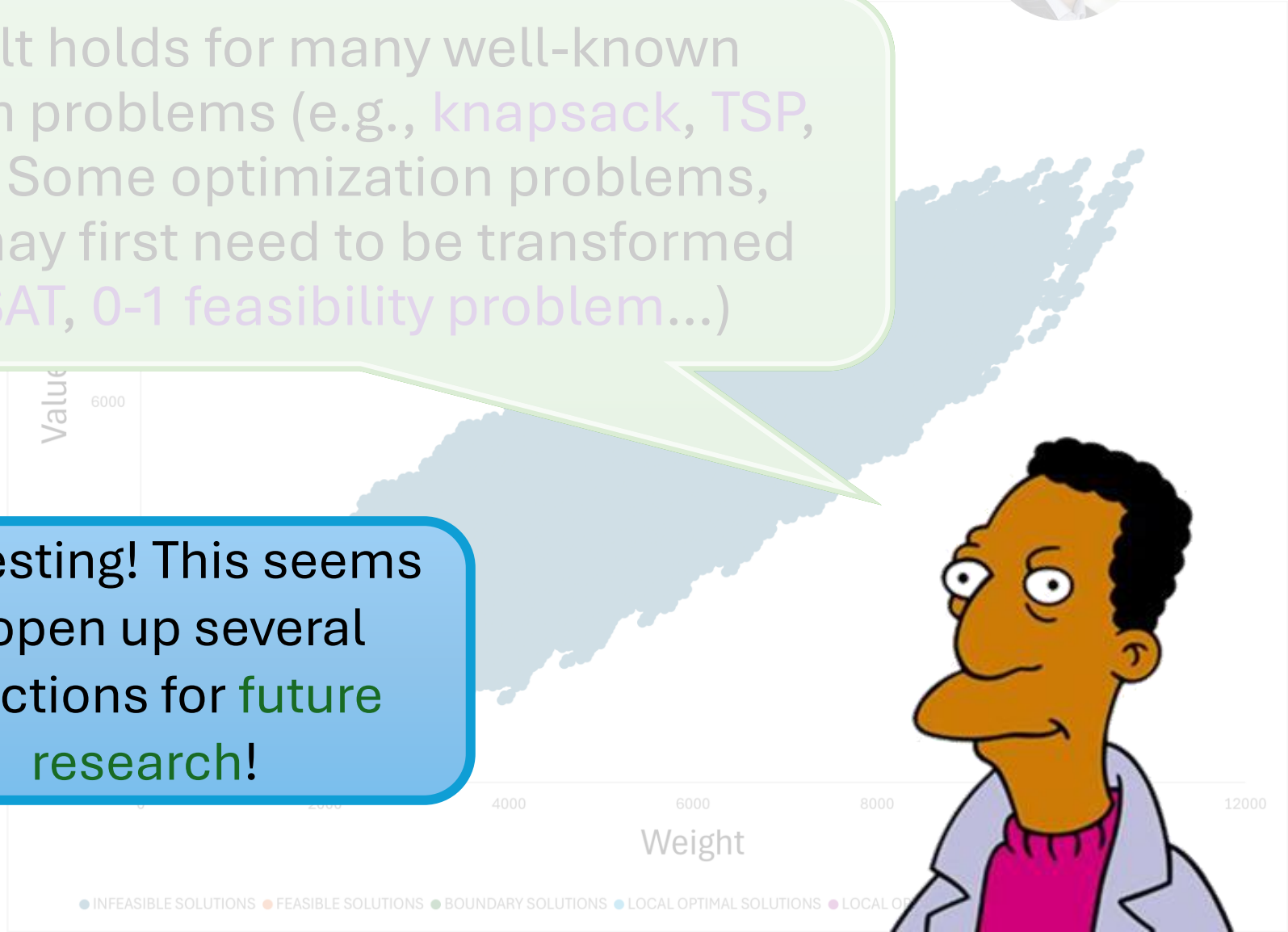


- If we use prob… dominance- a… rules, we can … and can obtai… locally optimal solutions.
- W… n obtain this …sition in polynomial …
- In … example there … gl… ally o… soluti… …se… tains t… …timal solutions).

This result holds for many well-known optimization problems (e.g., knapsack, TSP, RCPSP…). Some optimization problems, however, may first need to be transformed (e.g., 3-SAT, 0-1 feasibility problem…)

Interesting! This seems to open up several directions for future research!





# Some conclusions…

- As quantum algorithms are inherently probabilistic, it may be more challenging to obtain a linear speedup from parallel computing.
- Quantum algorithms may be used as general-purpose solvers, exact (hybrid) procedures, or heuristics.
- Compared to classical computing we can obtain up to a quadratic speedup when solving NP-Complete/Hard optimization problems. Even though these problems remain NP-Complete/Hard, a quadratic speedup is still very interesting from a practical point of view.
- Quantum computers excel at solving complex problems that have non-linear objective functions and/or constraints (in contrast to classical computers). Take for instance the quadratic knapsack problem.
- Once we have universal quantum computing, quantum algorithms will revolutionize the field of optimization!

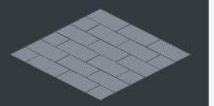
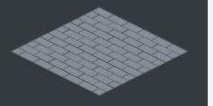
However, we may need to be patient for a bit longer…

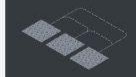
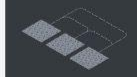
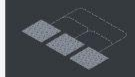
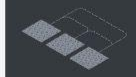
Email: sc@cromso.com

# Development Roadmap

IBM **Quantum**

| | 2016–2019 ✓ | 2020 ✓ | 2021 ✓ | 2022 ✓ | 2023 ✓ | 2024 | 2025 | 2026 | 2027 | 2028 | 2029 | 2033+ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Ran quantum circuits on the IBM Quantum Platform | Released multi-dimensional roadmap publicly with initial aim focused on scaling | Enhanced quantum execution speed by 100x with Qiskit Runtime | Brought dynamic circuits to unlock more computations | Enhanced quantum execution speed by 5x with Quantum Serverless and execution modes | Improve quantum circuit quality and speed to allow 5K gates with parametric circuits | Enhance quantum execution speed and parallelization with partitioning and quantum modularity | Improve quantum circuit quality to allow 7.5K gates | Improve quantum circuit quality to allow 10K gates | Improve quantum circuit quality to allow 15K gates | Improve quantum circuit quality to allow 100M gates | Beyond 2033, quantum-centric supercomputers will include 1000's of logical qubits unlocking the full power of quantum computing |
| Data scientists | | | | | | Platform | | | | | | |
| | | | | | | Code assistant | Functions | Mapping collections | Specific libraries | | | General purpose QC libraries |
| Researchers | | | | | Middleware | | | | | | | |
| | | | | | Quantum Serverless ✓ | Transpiler service | Resource management | Circuit knitting x p | Intelligent orchestration | | | Circuit libraries |
| Quantum physicists | | | Qiskit Runtime | | | | | | | | | |
| | IBM Quantum Experience ✓ | | QASM 3 ✓ | Dynamic circuits ✓ | Execution modes ✓ | Heron (5K) | Flamingo (5K) | Flamingo (7.5K) | Flamingo (10K) | Flamingo (15K) | Starling (100M) | Blue Jay (1B) |
| | **Early** ✓<br>Canary<br>5 qubits<br>Albatross<br>16 qubits<br>Penguin<br>20 qubits<br>Prototype<br>53 qubits | **Falcon** ✓<br>Benchmarking<br>27 qubits | | **Eagle** ✓<br>Benchmarking<br>127 qubits | | Error mitigation<br>5k gates<br>133 qubits<br>Classical modular<br>Up to 133x3 = 399 qubits | Error mitigation<br>5k gates<br>156 qubits<br>Quantum modular<br>Up to 156x7 = 1092 qubits | Error mitigation<br>7.5k gates<br>156 qubits<br>Quantum modular<br>Up to 156x7 = 1092 qubits | Error mitigation<br>10k gates<br>156 qubits<br>Quantum modular<br>Up to 156x7 = 1092 qubits | Error mitigation<br>15k gates<br>156 qubits<br>Quantum modular<br>Up to 156x7 = 1092 qubits | Error correction<br>100M gates<br>200 qubits<br>Error corrected modularity | Error correction<br>1B gates<br>2000 qubits<br>Error corrected modularity |

✓ Executed by IBM

On target