Solving Combinatorial Optimization and NP-Hard problems using novel Quantum Annealing using NEC Vector Engine

NEC Corporation

Combinatorial Optimization Problems

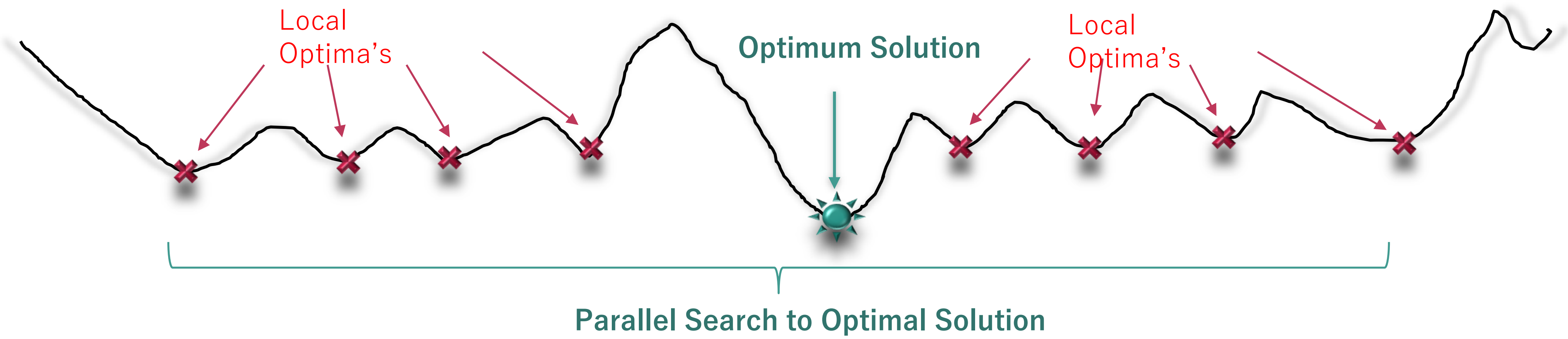
Combinatorial Problem

- Characterized by inputs:
  - an **objective** defining the properties of a solution
  - a general description of **conditions** and **parameters**
- Solved by:
  - Find a **group**, ordering, or assignment of a discrete, finite set of objects that **satisfies** given conditions.
- Combinations of objects or solution components that **need not** satisfy **all given conditions** are **candidate solutions**.
- Candidate solutions that satisfy **all given conditions** are the actual **solutions.**

Optimization Problem

- **Define** an objective function for the inputs.
- Objective function measures solution quality (often defined on all candidate solutions).
- **Minimize/Maximize** the objective function to find a solution with optimal quality.
- Variants of optimization problems:
  - **Search variant**: Find a solution with optimal objective function value for given problem instance.
  - **Evaluation variant**: Determine optimal objective function value for given problem instance.

Combinatorial Optimization Problems



- **Goal:** To find optimal solution or object from a finite set of solutions/space or objects.
- **Challenge:** The solution space is typically too large to search exhaustively using brute force or exploring multiple solution or many local minima's.
- **Examples:** Finding shortest/cheapest round trips(TSP), planning/scheduling, Supply Chain Optimization, Circuit design, Protein Structure Prediction etc.

Quantum Annealing (QA) can explore the solution space in parallel using energy fluctuations - Quantum Effect

Simulated Annealing (SA) simulates the same by a meta-heuristic approach of execution on a classical computer **using Vector Engine accelerators**

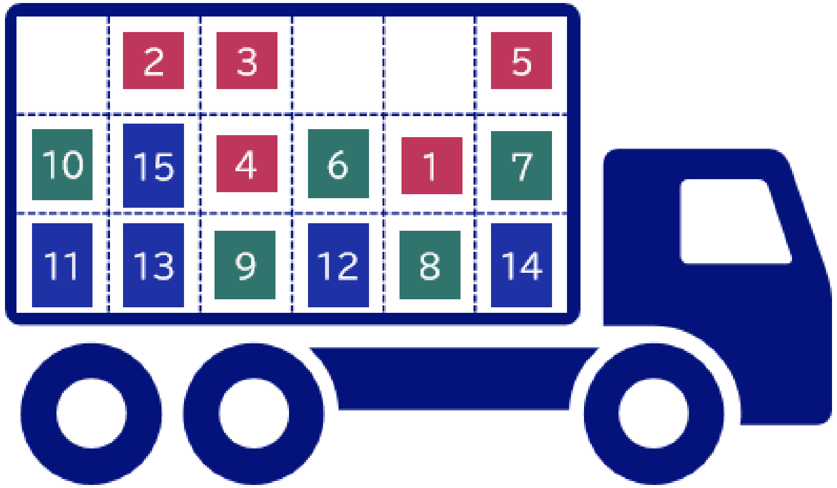
Use Case: Load-Limited Route Optimization Problem

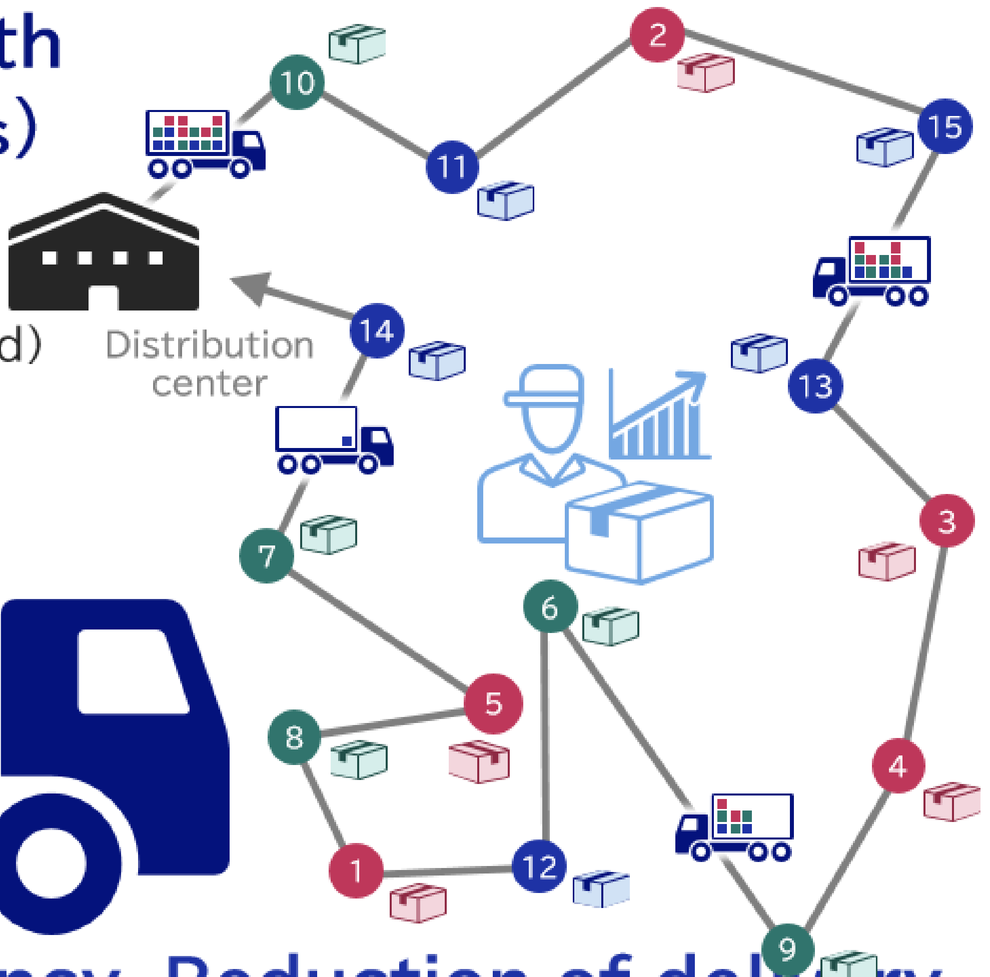
Route optimization problem by taking truck loading restrictions into account

Example: Route optimization problem with special constraints (e.g., loading restrictions)

- Capacity is 18 in 3 rows of 6
  (Cargo is removed from behind and above)
- 15 delivery locations (Allow free space in the cargo bed)
- Heavier cargo cannot be stacked on top

weight	quantity					
5kg	5	1	2	3	4	5
15kg	5	6	7	8	9	10
25kg	5	11	12	13	14	15





Effectiveness Improved delivery efficiency. Reduction of delivery time Reduced burden on delivery staff

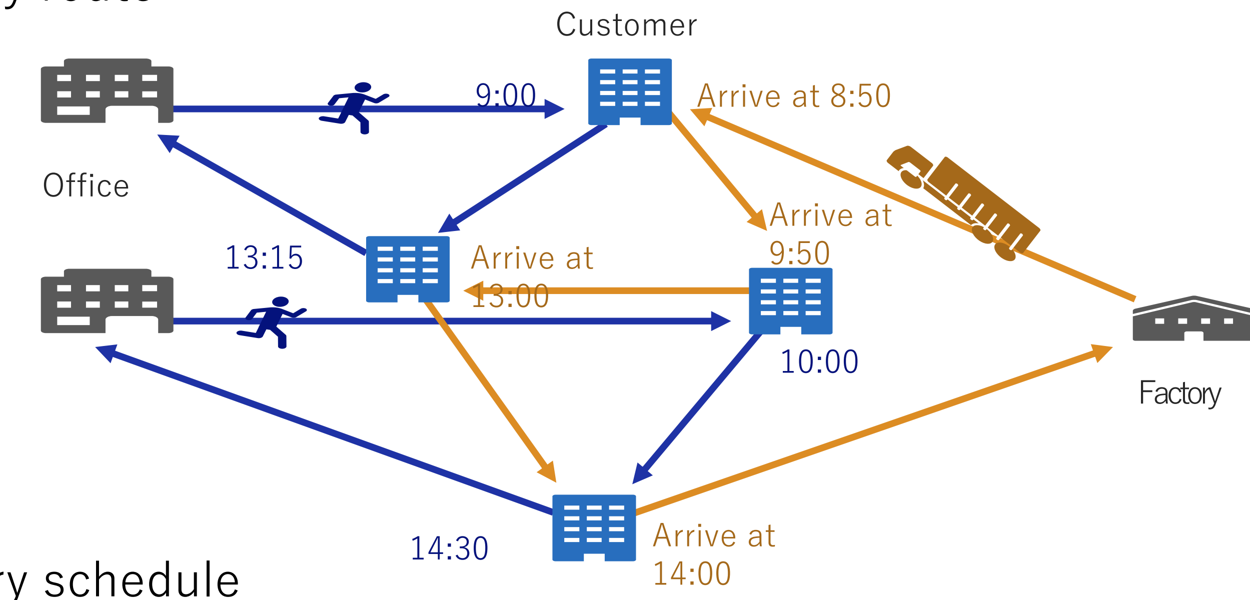
Use Case: **Delivery Route and Schedule Optimization**

for reducing costs, time, energy, CO_2, etc.

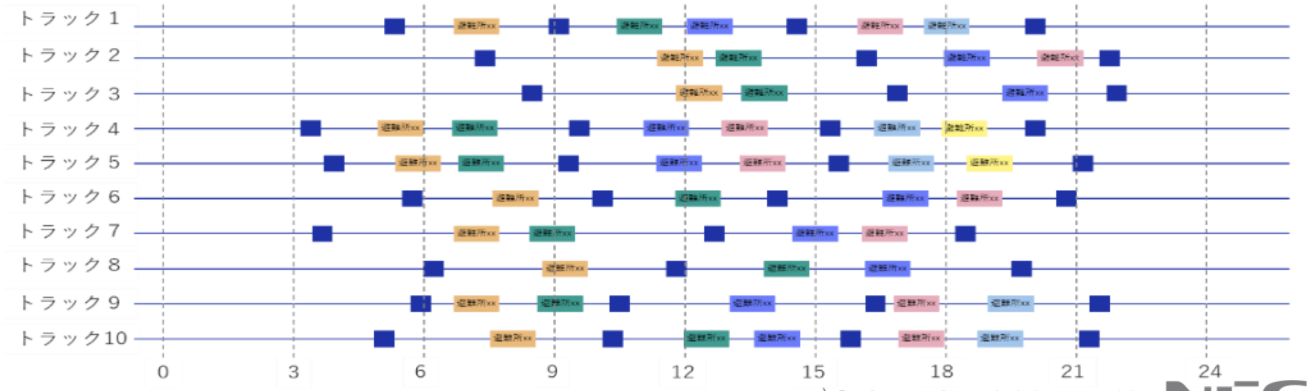
EX. Delivery of parts and dispatch of Engineers

- Parts are delivered by truck
- Engineers move by car/train
- Have to consider skills of each engineer

delivery route

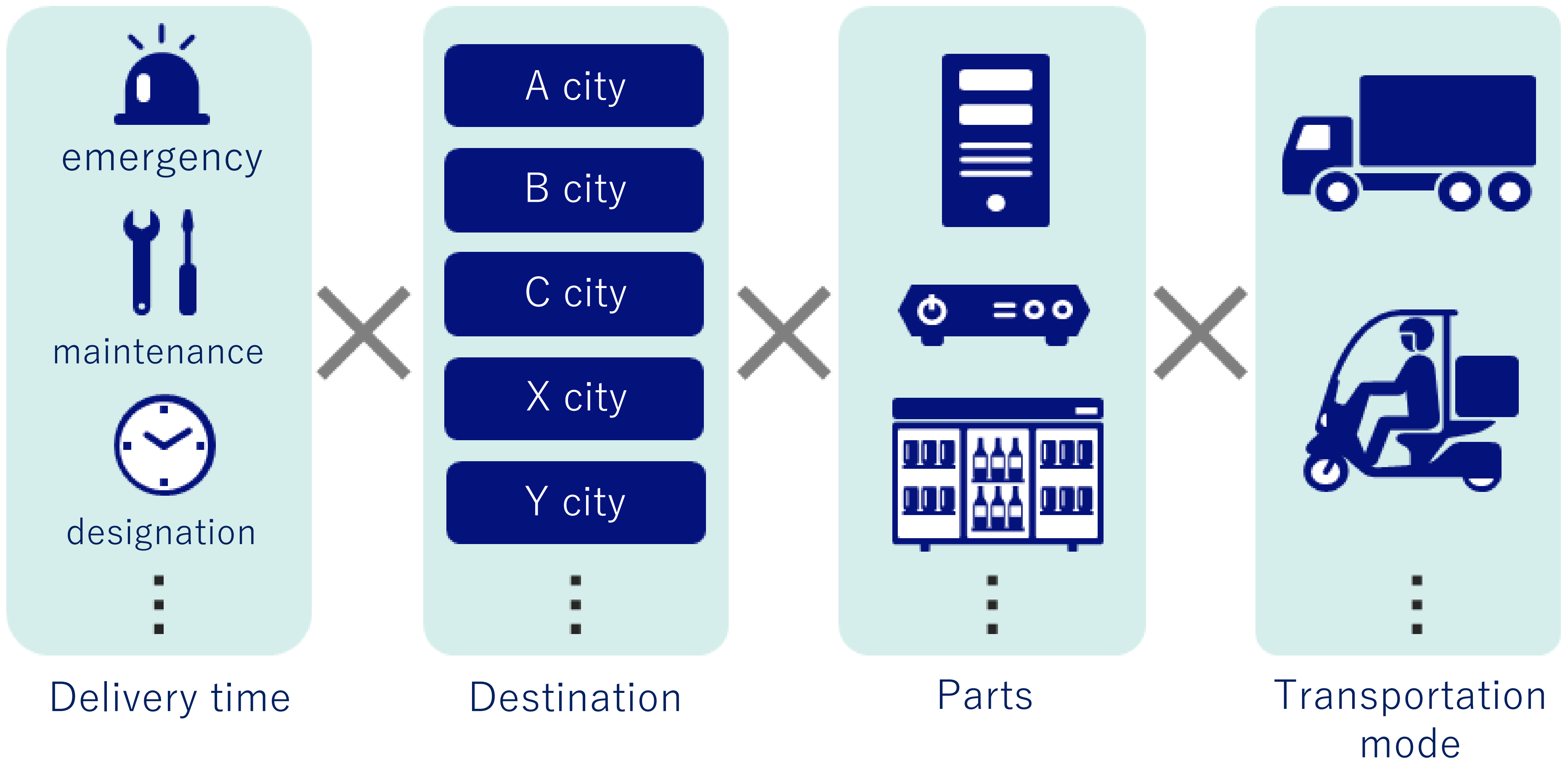


delivery schedule



\Orchestrating a brighter world NEC

Combinatorial optimization from complex combinations



Actual Operations by VA

Applying VA optimization to delivery orders from a day before

In actual operation since Oct 2022

Delivery orders the day before

Input

Several 100s of orders /day

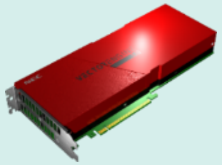
Professional engineer optimization

120 min./day

Delivery Planning

- ・Delivery time
- ・Traffic Jam
- ・Delivery area
- ・Parts count
- ・Parts weight
- ・Transportation mode

Maintenance at customer



Vector Annealing optimization

12 min./day

Future Plan

Delivery order same day and Emergency orders

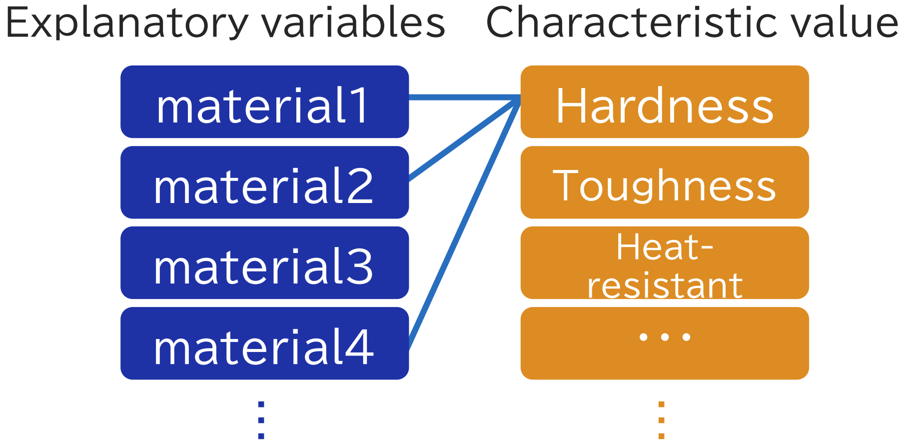
30% cost reduction

CO_2 reduction

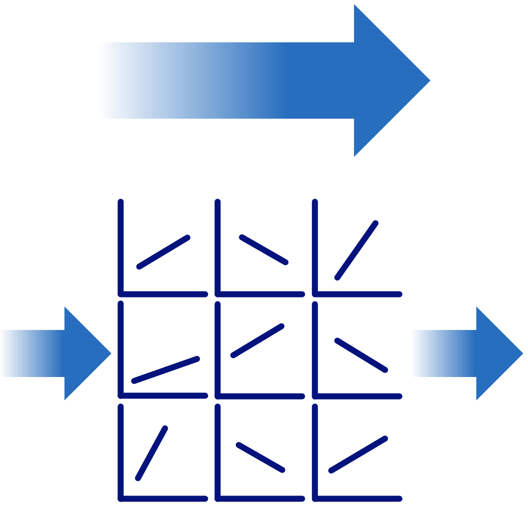
Towards advanced MI (Materials Informatics) key technology

- Predictive model creation by AI
  → Quantum computing solves the **inverse problem**
- Providing **multiple influential recipe** proposals

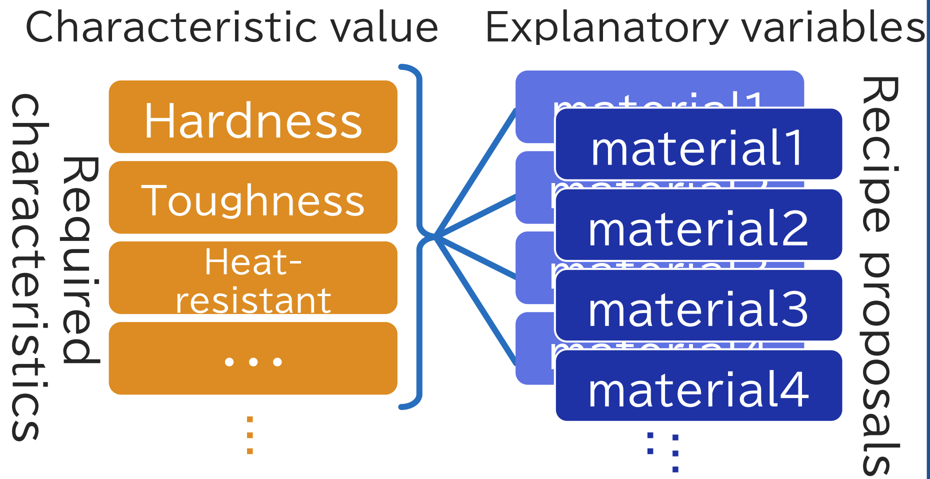
①Predictive model by AI



Create a predictive model of the impact of material type and quantity on product characteristics from past experimental data



②QC solves the inverse problem



Find the optimal combination of the amount of material that expresses the characteristic target value

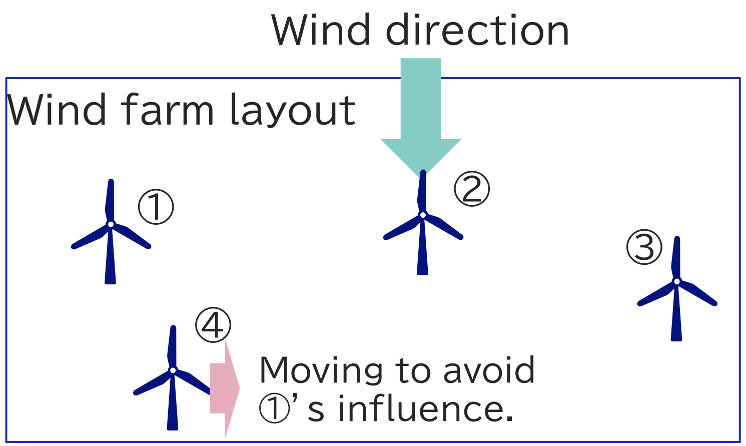
Wind farm layout optimization

Confirmed quantum computing feasibility with original calculation model.

※ https://www.ctc-g.co.jp/company/release/20220616-01445.html (Japanese Only)

Evaluation of Wind-Power Generation

- The electric power generation is changed by natural conditions.
- The amount of power generation is changed by wind farm layout due to each windmill interaction.



The electric power generation change.

Calculation model

- 10,000 installation location candidate in 12km x 12km region
- 20 windmills placement is optimized

Conventional simulation

10 hours calculation time

- The time increase exponentially by increasing installation region and windmills.

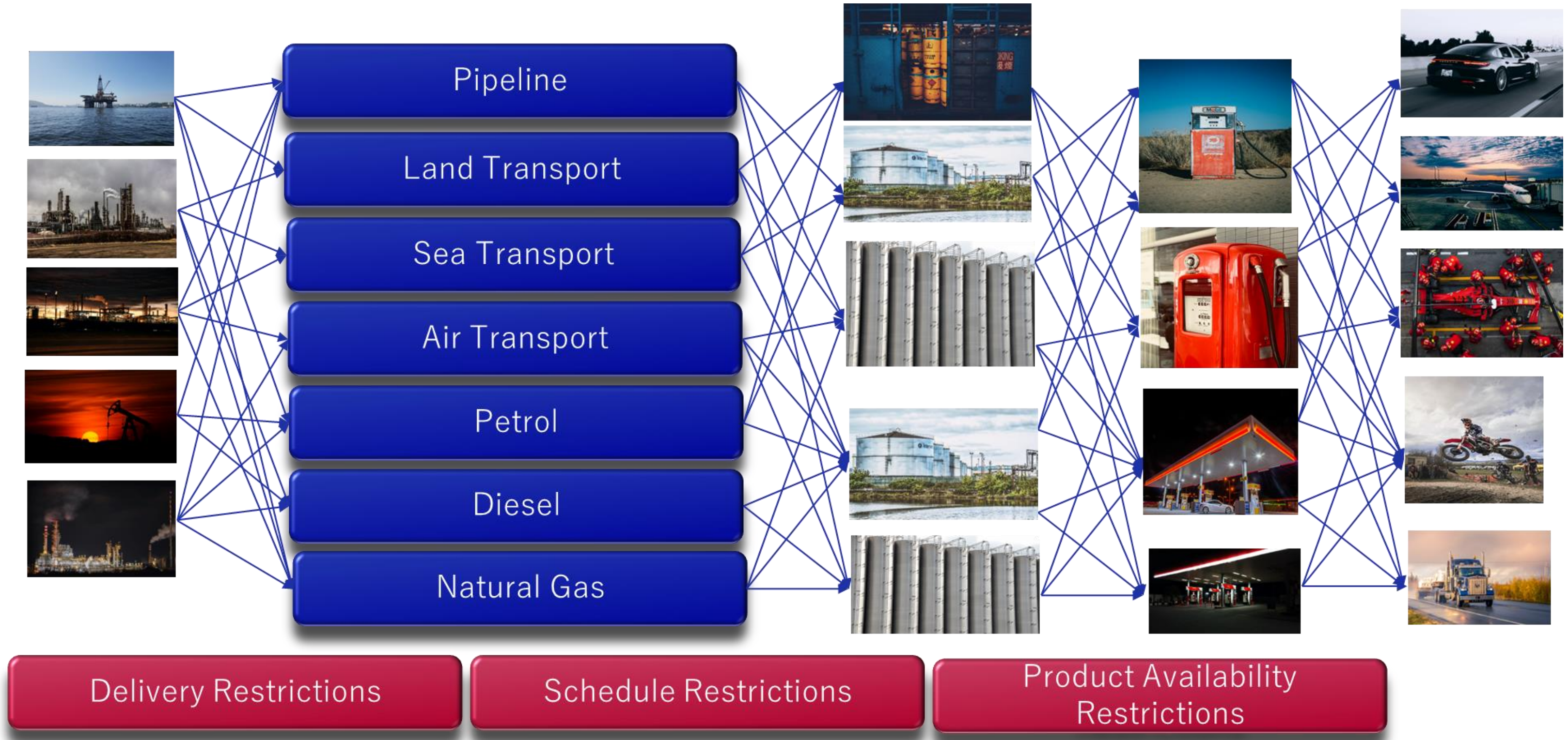
Calculation by Vector Annealing

Only 10 min calculation time.

- The result is equivalent with conventional simulation result.

Optimizing Supply Chain Optimization for delivering Energy

- **Requirement**: Reduce or efficiently manage the **cost** of transportation for the downstream supply-chain.
- **Challenge:** massive number of **combinations of paths** or **routes** to choose from along with various **restrictions** in a real-world scenario.



an NP-Hard problem or combinatorial optimization problem!

Annealing

- “Annealing” is the process of heating a material to critical temperature levels resulting in structural or property changes, followed by cooling it to retain the change.

  Example: *Forging swords*.

- Simulated Annealing (SA) is a probabilistic, metaheuristic technique inspired from process of annealing metals for solving optimization problems.

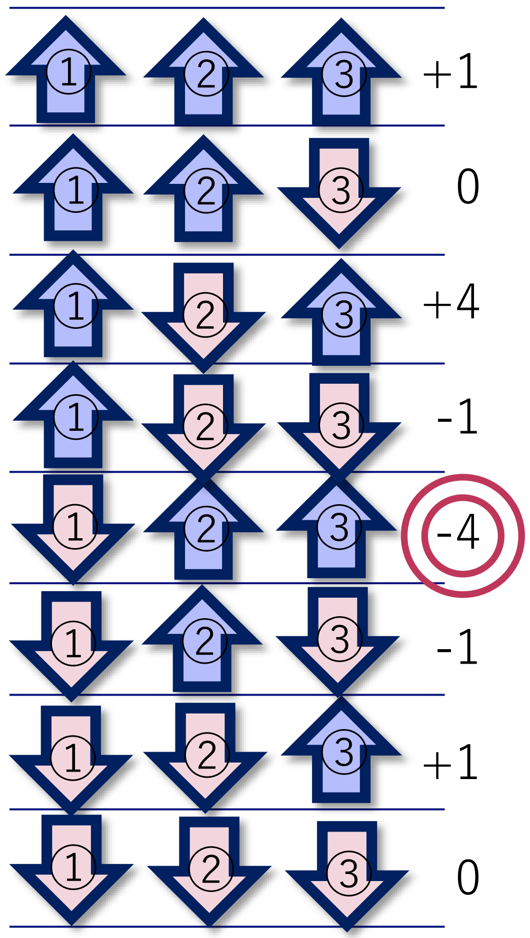
- The goal is to achieve minimum energy (entropy), or temperature that results in high probability of success of achieving a speedy as well as accurate solution!

Simulated Annealing

- A variable can be denoted as taking values of either **TRUE (0)** or **FALSE (1)**, *or* denoted as magnetic spins **spin up (↑)** and **spin down (↓)**

- If the number of spins is 3, the optimal one is obtained with 2^3, or 2 x 2 x 2 = 8 combinations in a single annealing process.
  - If 10 spins, $2^{10}=1024$
  - If 20 spins, $2^{20}≒1$ million
  - If 30 spins, $2^{30}≒1$ billion
  - If 1000 spins, $2^{1000} ≒ 1.071509 \times 10^{301}$

We can find a good combination out of a huge number of combinations!

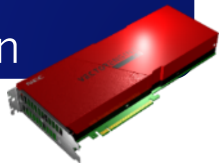


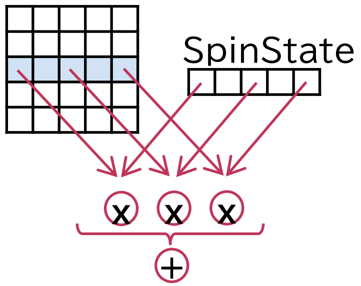
Vector Annealing

6x Acceleration by the Vector Architecture

50x Acceleration by skipping constraint violations

Vector operation on VE
Energy calculation is matrix operation



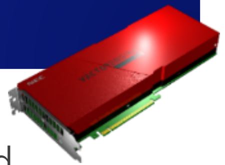



```
for(i=0; i<numSpins; i++)
  DeltaEnergy +=
        Qij[FlipSpin][i] x SpinState[i]
```

Effect from neighbor spin

0 or 1

Full connect 100k bits/VE and high memory bandwidth

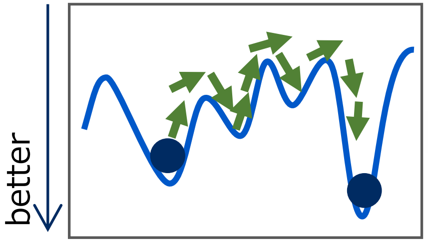


- 48GB memory capacity and 1.5TB/s memory bandwidth
- Multi card supports larger number of qbits (100k qbits x n)$^{1/2}$

Optimized algorithm for VE
Avoiding Redundant Search

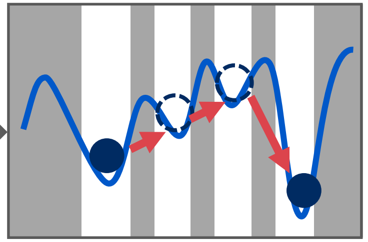
Existing search

Including constraint violations



VA search

skip constraint violations



Problem

Formalize

Search

Check constraint

Violation
→try again

Problem

Formalize

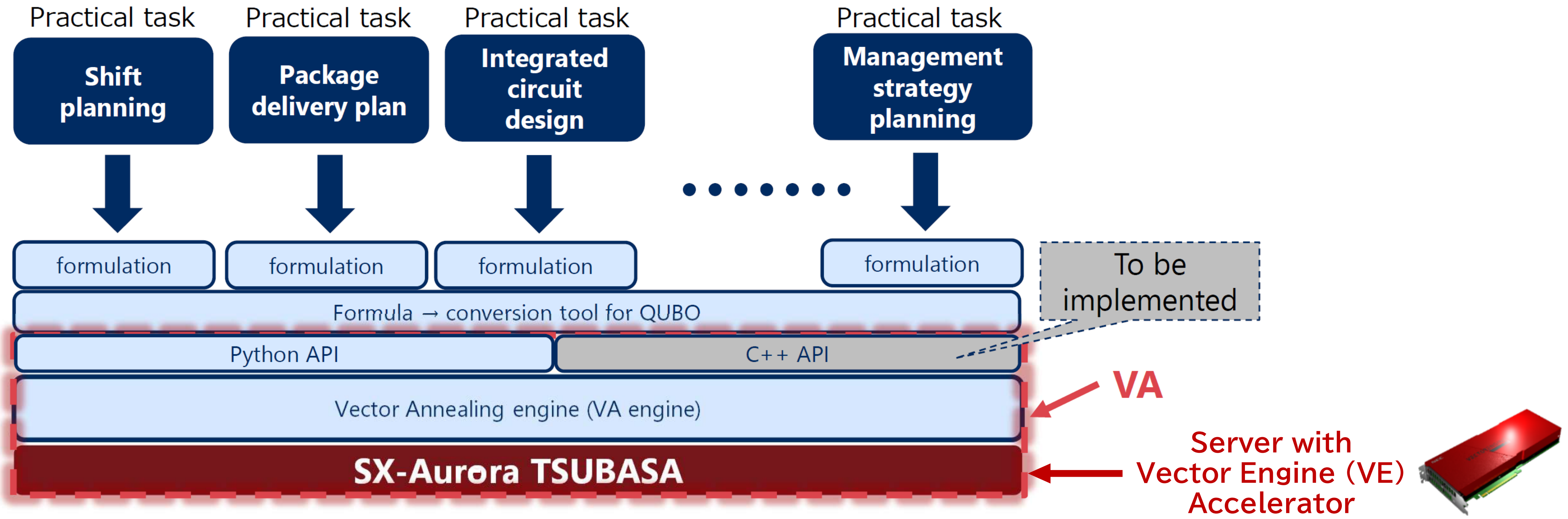
Search considering constraint

Constraint

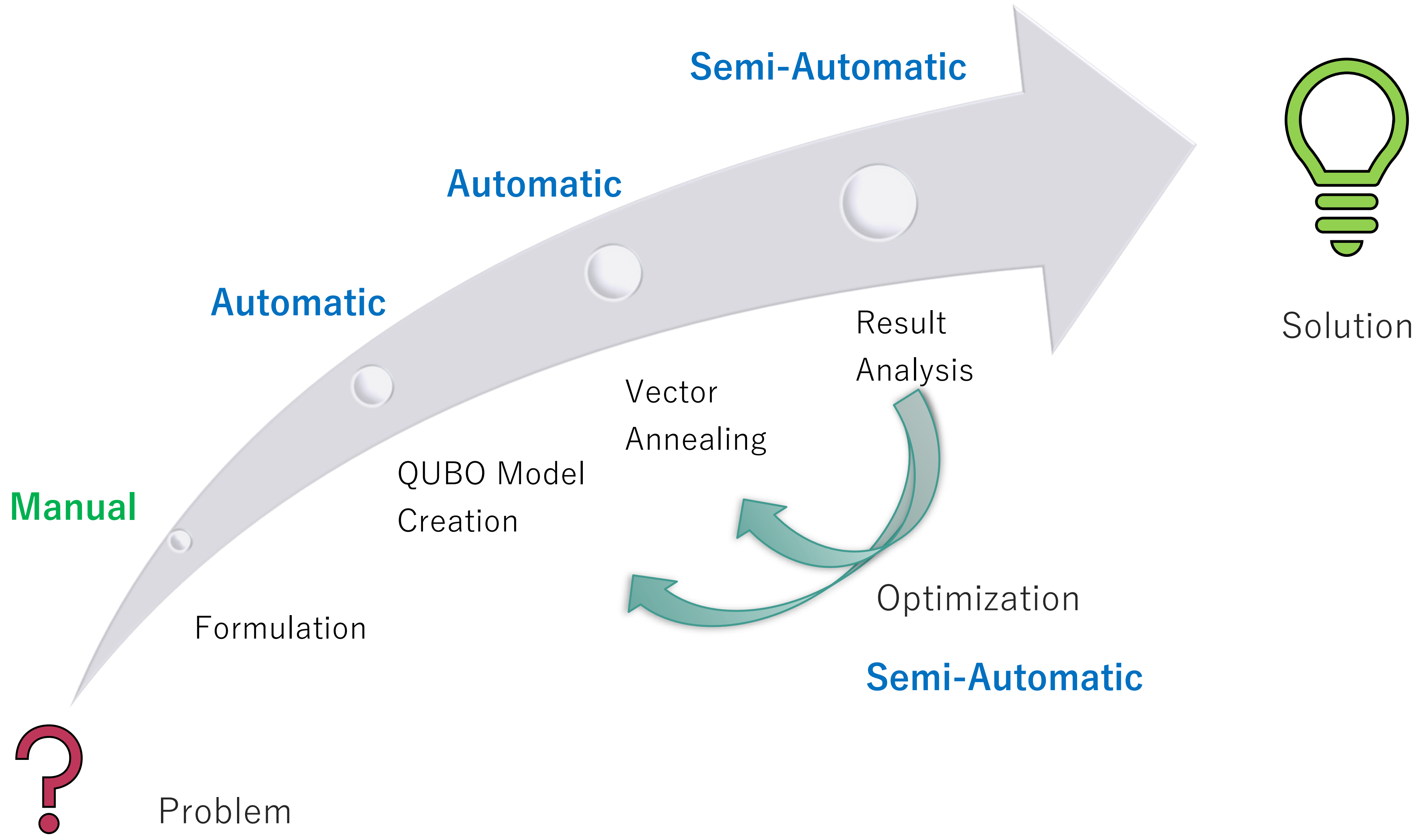
computational complexity reduction

Vector Annealing Solution Stack

Input	QUBO format
Problem Size	Up to 300K Qubit/Variables 8 x Vector Engine cards
Connection	32-bit floating point, full connection
Algorithm	Includes our extension to improve result



Solution approach for Vector Annealing



Problem Formulation

- **Hd (Objective Function):** Minimize the distance of the delivery route.

- **H1 (Constraint):** Visit a certain location twice or even three times because there are three extra available boxes. (for example: 18 boxes for 15 locations).
  Therefore, the Hamiltonian is created so that the energy is lower if the order bit stands 1 to 3 times for each location.

- **H2 (Constraint):** Since multiple packages cannot be delivered to the same point, only one location can be visited at a time.

- **H3 (Constraint):** Penalize when the heaviest luggage comes on top.

- **H4 (Constraint):** Locations that need to be accessed twice or three times should be consecutive.

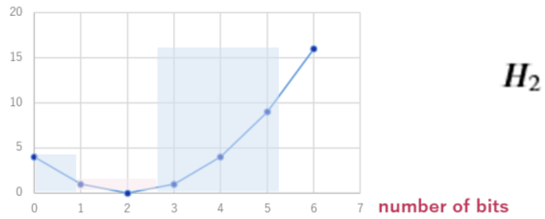
$$H = H_d + \beta_1 * H_1 + \beta_2 * H_2 + \beta_3 * H_3 + \beta_4 * H_4$$

Initially, all the weights are kept small, and the weights are adjusted by trial and error, such as making the weights stronger for constraints that are not satisfied.

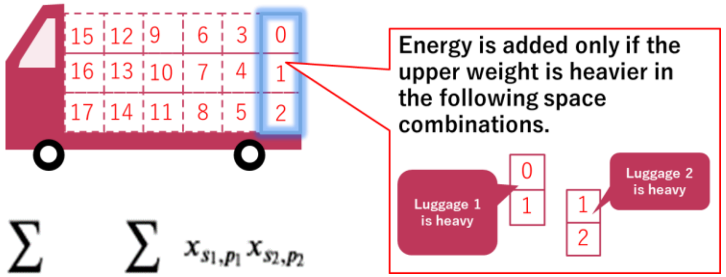
$$H_d = \overbrace{\sum_{p\in P} d_{C,p} x_{0,p}}^{\text{Distance from the center}} + \overbrace{\sum_{i=0}^{capa-2} \sum_{p_1,p_2\in Q} d_{p_1,p_2} x_{i,p_1} x_{i+1,p_2}}^{\text{Add up the distance between i and the i+1st delivery destination}} + \overbrace{\sum_{p\in P} d_{p,C} x_{capa-1,p}}^{\text{Distance back to the center}}$$

$$H_1 = \sum_{p\in P} \left(\sum_{i=0}^{capa-1} x_{i,p} - 2 \right)^2$$

If the order bit is high twice, the energy is 0.
If the order bit high once or three times, the energy is 1.
Otherwise, the energy increases exponentially as you move away from 2.



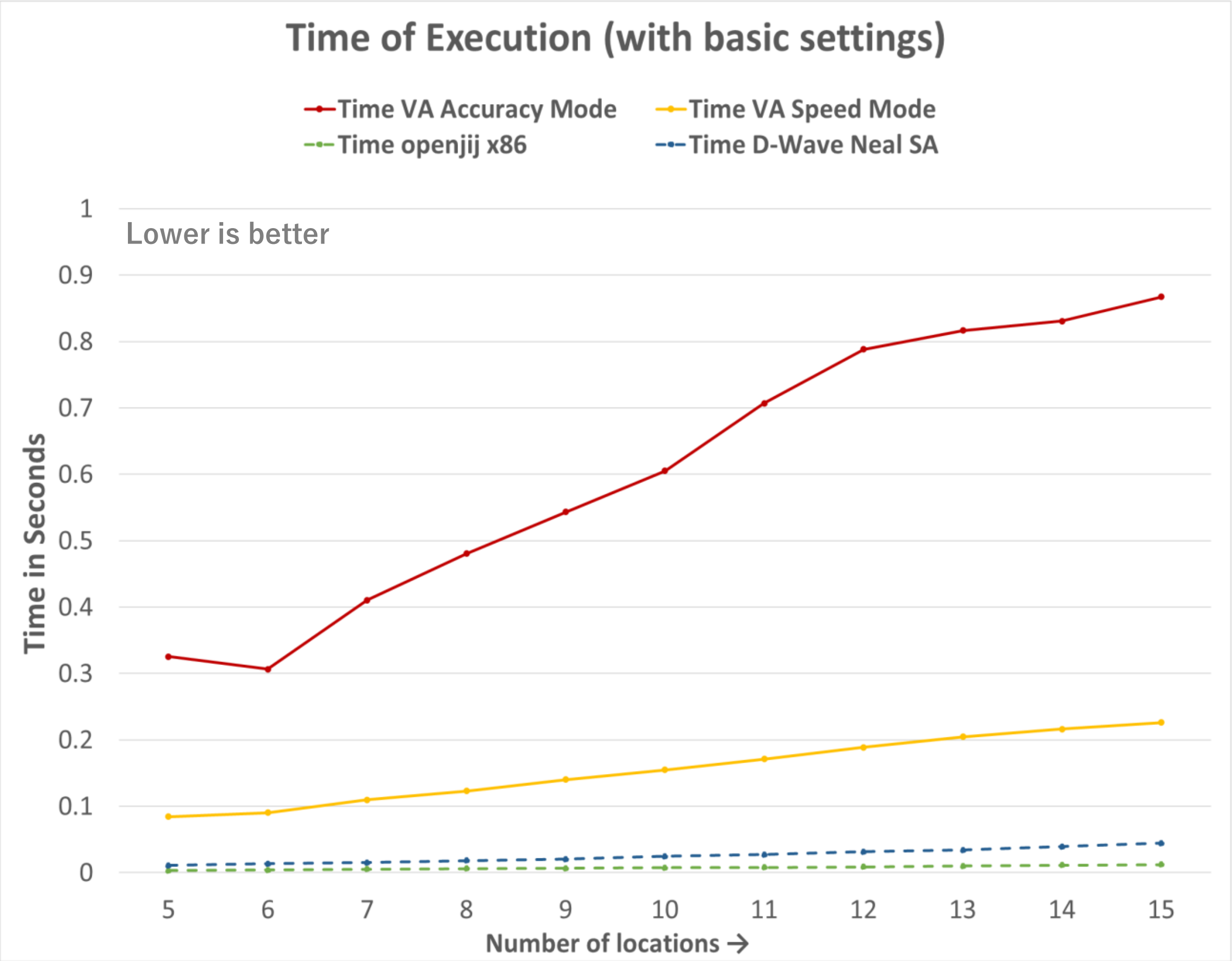
$$H_2 = \sum_{i=0}^{capa-1} \left(\sum_{p\in P} x_{i,p} - 1 \right)^2$$

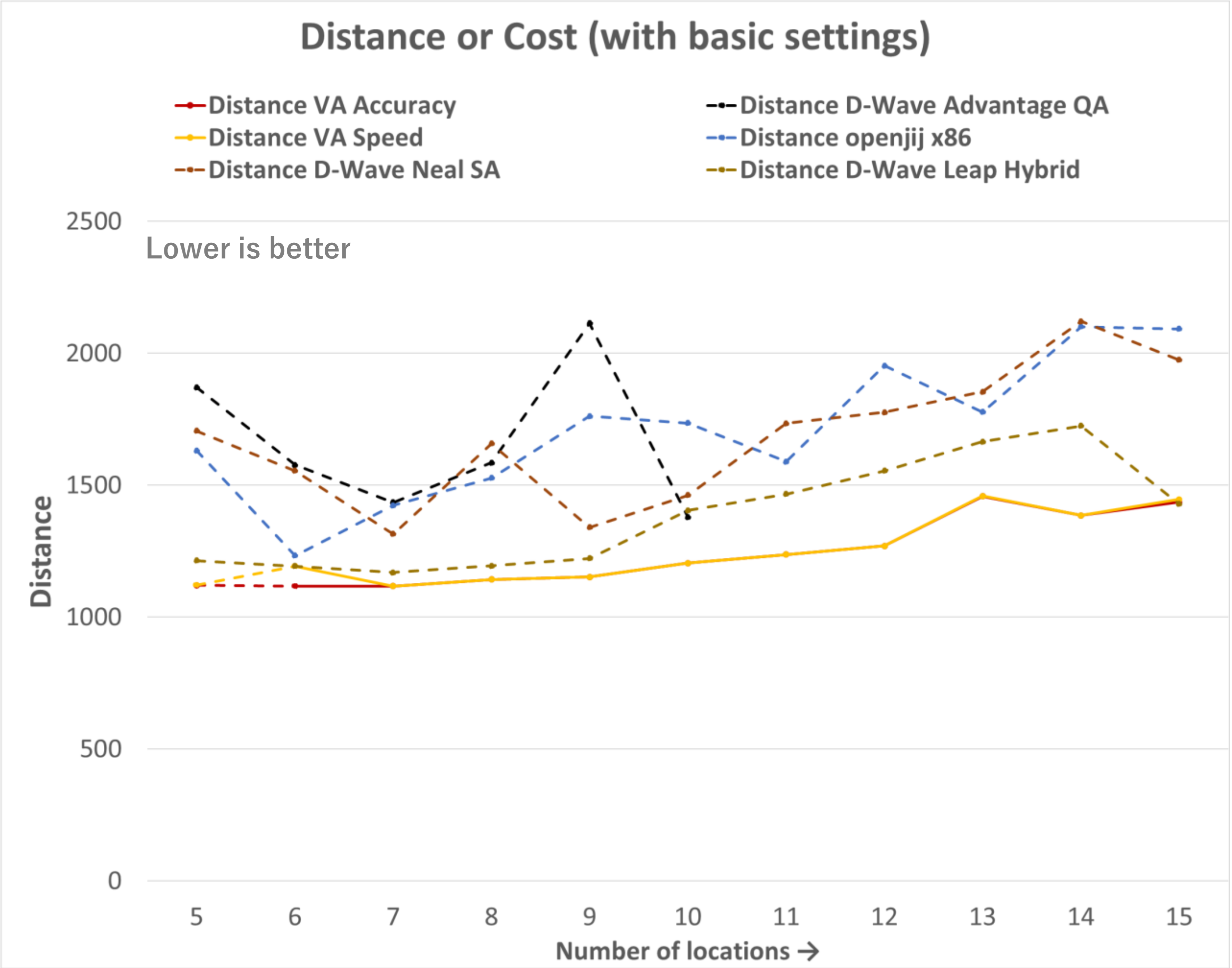


$$H_3 = \sum_{s_1,s_2\in space\_pair} \sum_{p_1,p_2\in R} x_{s_1,p_1} x_{s_2,p_2}$$

$$H_4 = -1 * \sum_{p\in P} \left(\sum_{s_1,s_2\in space\_pair} x_{s_1,p} x_{s_2,p} \right)^2$$

Result Analysis (15 locations dataset)



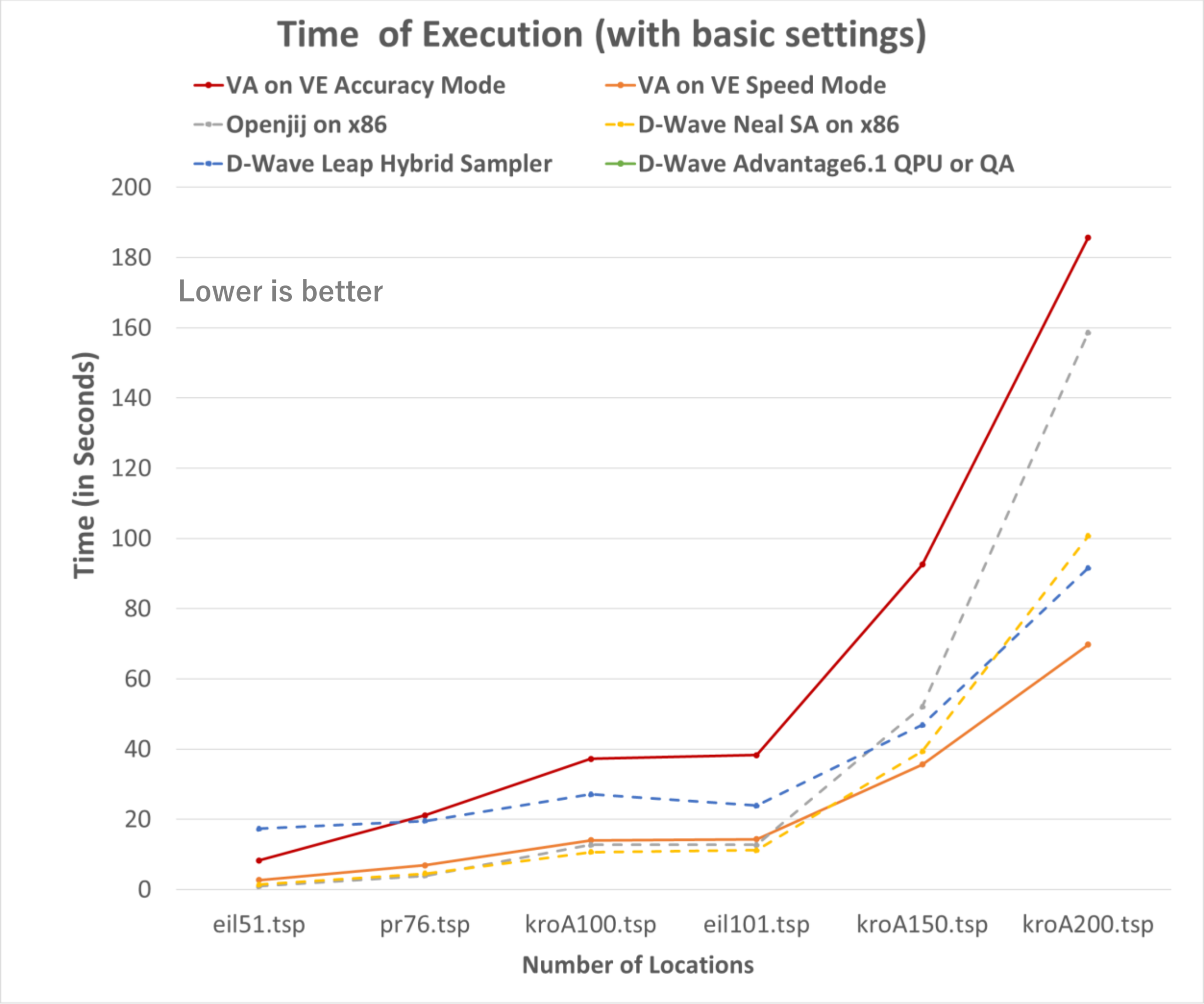


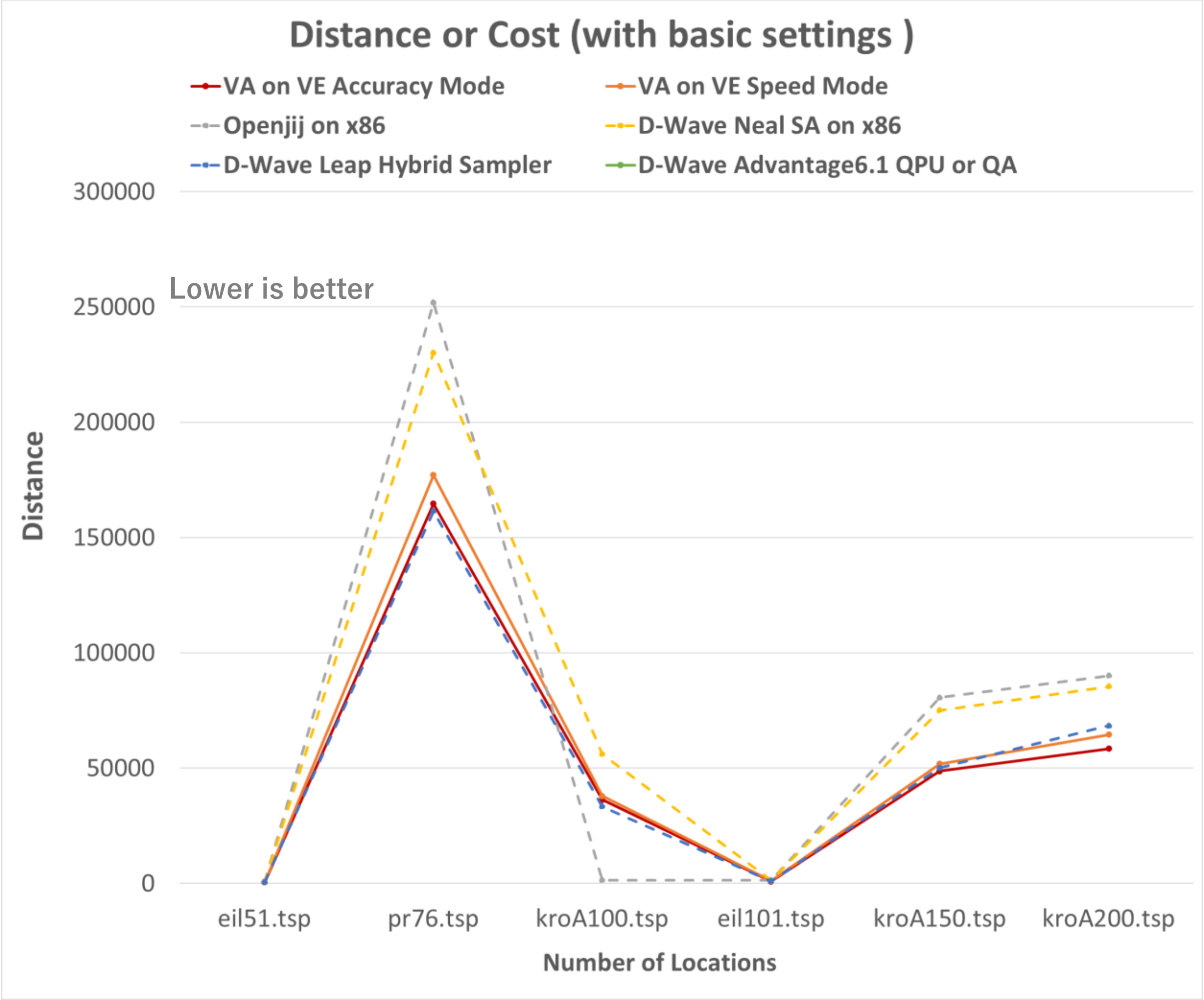
QA and Leap Hybrid Time is not included, because it was cloud access, and it had a large value than 1 second.

- **Dotted Lines represent the constraints were broken by the solver to get the results.**
- **Solid Lines represent no constraints were broken.**

Result Analysis (51 to 200 locations public dataset)

No Hardware embedding found within a time limit due to free cloud access only



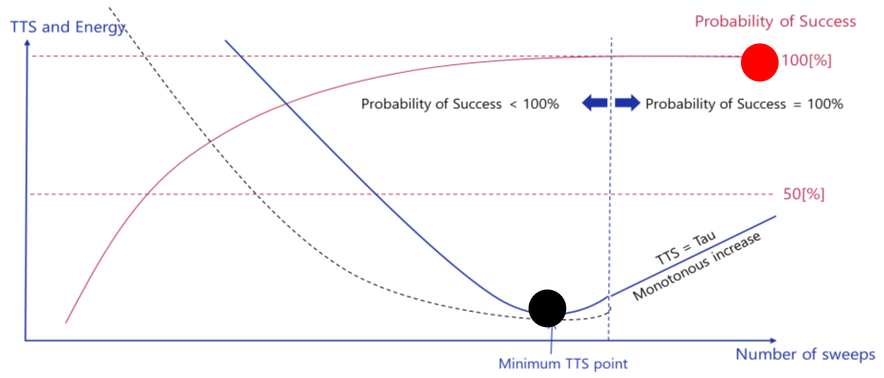


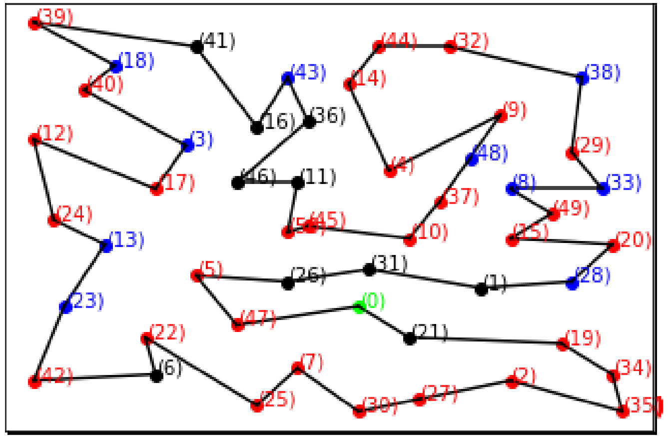
- **Dotted Lines represent the constraints were broken by the solver to get the results.**
- **Solid Lines represent no constraints were broken.**

Source of data: http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/tsp/

Result Analysis (with Sweeps Optimization)

- The results obtained while using public dataset eil51.tsp or 51 locations results considering real-world constraints along with minimum distance achievement.
- Results for speed focus (try to use as minimum sweeps as possible):
  - Best Minimum Distance: **501**
  - Best Minimum Energy: **62**
  - Single Time of Execution: **2.719 seconds**
- Results for accuracy focus (try to use as maximum sweeps as possible):
  - Best Minimum Distance: **478**
  - Best Minimum Energy: **61**
  - Average Time of Execution: **47.837 seconds**
- Increasing the number of sweeps optimization could reach better optimized solution.
- Time of execution can be further optimized using automated parameter estimators for coefficient values and using by lesser number of sweeps.





Sweeps = 12000 Distance = 478

Conclusion

- Simulated Annealing or Vector Annealing can be used to solve complex, real-world, multiple optimization problems which would benefit like demonstrated today for Oil and Gas supply chain optimization problem by cost of transport and CO2 emissions reductions.

- The python codes developed using SA or VA can be re-used/worked together in hybrid for an actual Quantum Annealing system in the future.

- NEC continues to work on solving more combinatorial optimization problems like knapsack, scheduling, manufacturing processes etc. to provide value to the workflow of oil and gas and other verticals.

References

- https://www.youtube.com/watch?v=mvCh6wgFSlk&t=1216s
- https://www.forbes.com/sites/paulnoble/2022/07/27/energy-oil-and-gas-the-network-visibility-imperative/?sh=11b2a8d9202d
- https://www.forbes.com/sites/muqsitashraf/2022/05/09/the-energy-industrys-supply-chain-dilemma-collective-pain-calls-for-collective-action/?sh=5702a80ed123
- https://www.nec.com/en/press/202208/global_20220829_01.html

\Orchestrating a brighter world

NEC