Large-Scale AGV Routing Based on Multi-FPGA SQA Acceleration

ASP-DAC 2025, Jan 20–23, 2025, Tokyo, Japan

Nguyen Quang Thinh

Thinh Nguyen Quang
Sharp Corporation,
Yamatokoriyama, Nara, Japan

Kosuke Matsuyama
Sharp Corporation,
Yamatokoriyama, Nara, Japan

Keisuke Shimizu
Sharp Corporation,
Yamatokoriyama, Nara, Japan

Hiroki Sugano
Sharp Corporation,
Yamatokoriyama, Nara, Japan

Eiji Kurimoto
Sharp Corporation,
Yamatokoriyama, Nara, Japan

Hasitha Muthumala Waidyasooriya
Graduate School of
Information,Tohoku University,
Sendai, Miyagi, Japan

Masanori Hariyama
Graduate School of
Information,Tohoku University,
Sendai, Miyagi, Japan

Masayuki Ohzeki
Graduate School of
Information,Tohoku University,
Sendai, Miyagi, Japan

Outline

1. **Introduction:** Problems, and objective

2. **Related work:** Application research

3. **Problem formulation:** model real-world large-scale AGV Operation system

4. **Proposed solution:** multi-FPGA SQA accelerator and system architecture

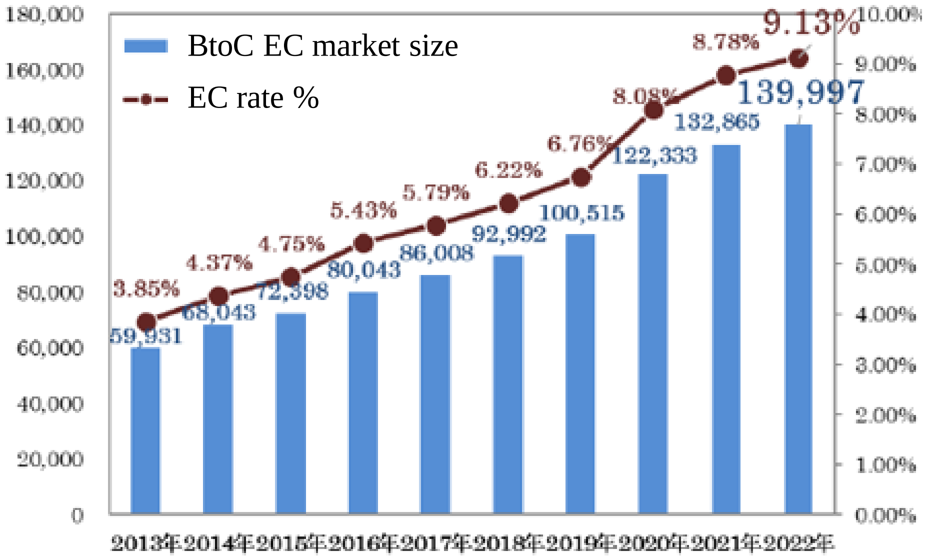
5. **Evaluation:** results and discussion

6. **Conclusion:** Key takeaways and future work

1. Introduction: Social and logistic problems in Japan

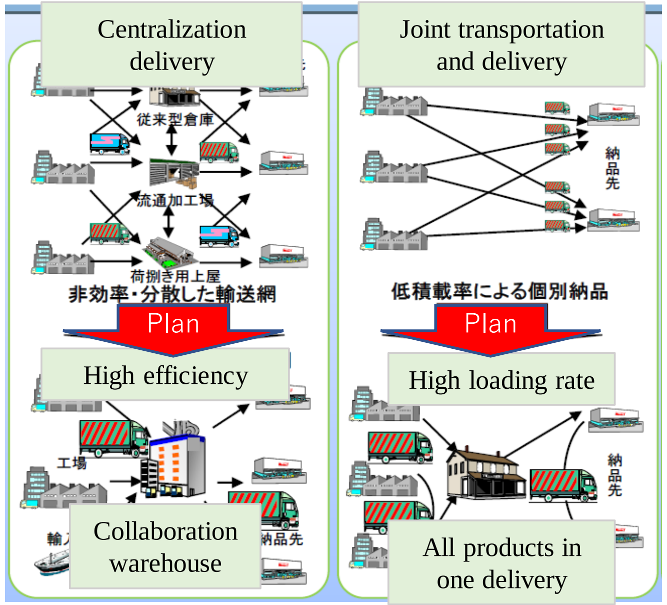
SHARP
Be Original.

The rapid growth of the e-commerce (EC) market



Source: The market size of the product-based e-commerce sector and the trend in e-commerce penetration rate (2022, Ministry of Economy, Trade, and Industry E-commerce Data)

Collaboration among logistic providers



Source: Working Group on Achieving Sustainable Logistics (Ministry of Economy, Transport and Tourism, Ministry of Agriculture, Forestry and Fisheries)

【Social problems】

The rapid growth of the EC market in Japan face subproblems:

- Delays of logistics infrastructure, requires significant time and cost.
- Labor shortages in Japan.
- The need to establish working environments.

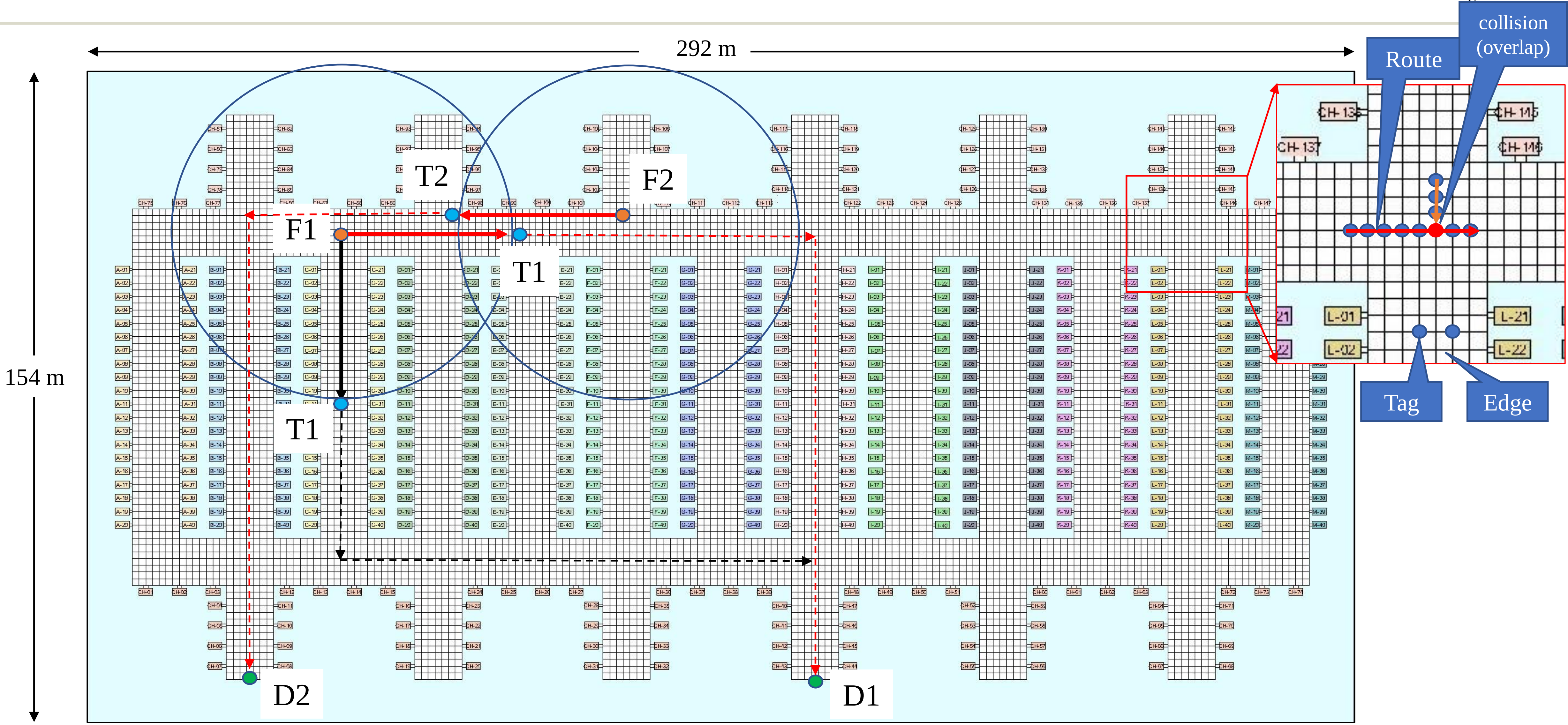
【Logistic problems 】

Logistic provider need collaboration delivery due to constrains:

- Build efficient delivery systems is difficult.
- Implement automation systems is high cost.
- The cost burden required for appropriate working environments.

Requires more and more the large-scale facilities and advanced automation of the warehouse system.

1. Introduction: Basic Route Optimization Problem for Mobile robots (AGV) in Large scale Warehouse



【Conditions】

-Number of AGVs: 1,000;

-Point of positions: 8,424 Tags

【Route Optimization】

-Route optimization process time depend on the complexity (number of AGVs, route requirements…)

-Real application requires safe and efficiency control method

2. Related work: Preview research

[M. Ohzeki, et.al., 2019]

Object function: Maximize the total length of the routes employed by each AGV

$$-\sum_{i=1}^{N}\sum_{\mu\in M(x_i,s_i)} d_{\mu}q_{\mu,i}$$

Constraint 1: 1 AGV select 1 route

$$+\lambda_1\sum_{i=1}^{N}\left(\sum_{\mu\in M(x_i,s_i)} q_{\mu,i}-1\right)^2$$

Constraint 2: No route overlap as much as possible within the estimated time T

$$+\lambda_2\sum_{e\in E}\sum_{t=1}^{T}\left(\sum_{i=1}^{N}\sum_{\mu\in M(x_i,s_i)} F_{\mu,t,e}q_{\mu,i}-1\right)^2$$

Problem scale simulation: ~10 AGVs, Speed =0.5 m/s

[R. Haba, et.al., 2022]

Object function: Minimize the total remain length of the routes employed by each AGV

$$\sum_{i=1}^{N}\sum_{j=1}^{M_i} d^{*}_{i,j}q_{i,j}$$

Constraint 1: 1 AGV select 1 route

$$+\,a\sum_{i=1}^{N}\left(\sum_{j=1}^{M_i} q_{i,j}-1\right)^2$$

Constraint 2: No route overlap as much as possible within the estimated time T

$$+\,b\sum_{e\in E}\sum_{t=1}^{T}\left(\sum_{i=1}^{N}\sum_{j=1}^{M_i} F_{i,j,t,e}q_{i,j}-\frac{1}{2}\right)^2$$

Problem scale simulation: ~20 AGVs, Speed =0.5 m/s

Difficult to apply to real world routing optimization of large-scale AGV control systems

2. Related work: Target problems and objectives

Input/output stations: 12

Products:
0.5~1.5 M

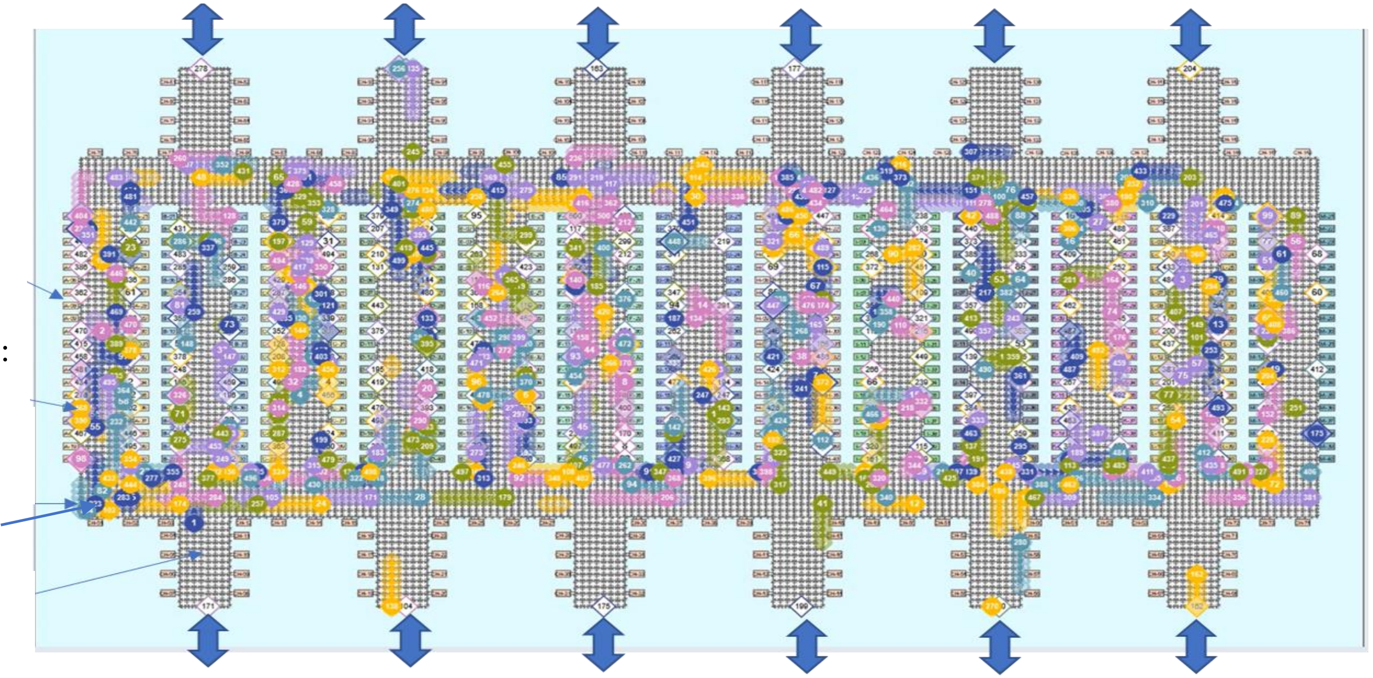
Pick Station:
520

Mobile robots:
1,000

Max speed:
1.67 m/s

Routes

Tags:
8,424



Overview of Hypothesis Testing for Route Optimization

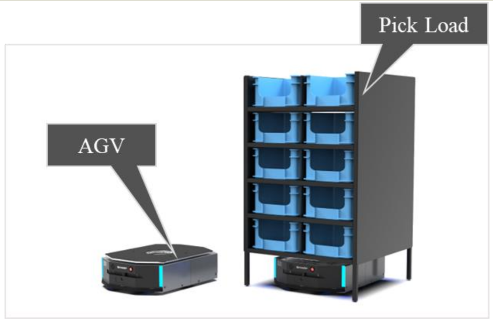
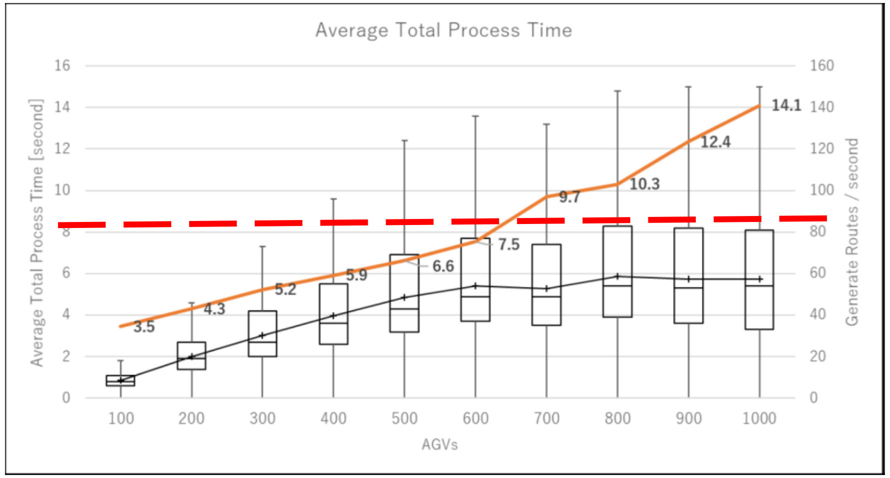


Figure 1:An AGV used in large logistics warehouses[7]

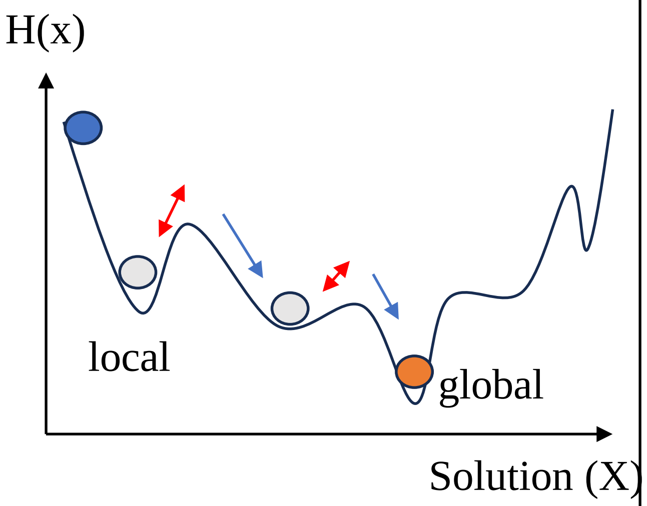
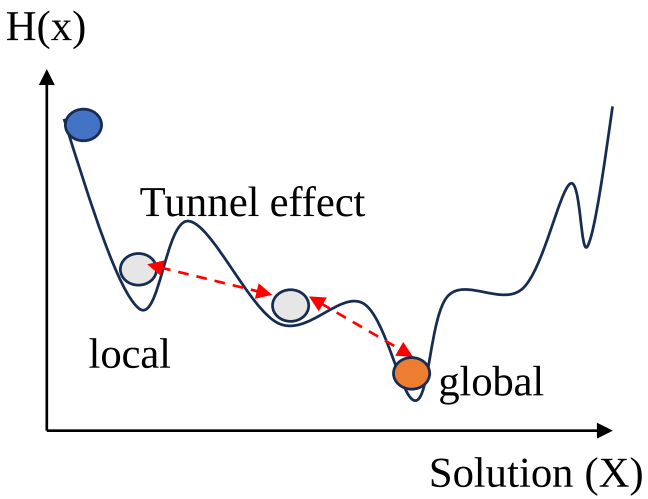
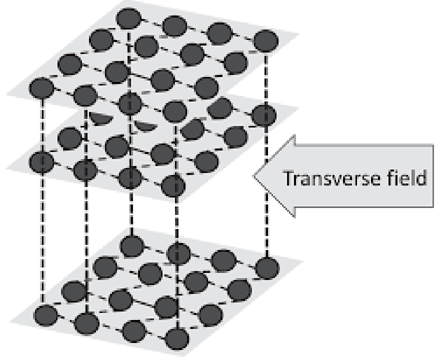


Current AGV control system performances
[N.Q Thinh, et.al., 2024]

The route optimization in large-scale warehouses is not only large but also complex.

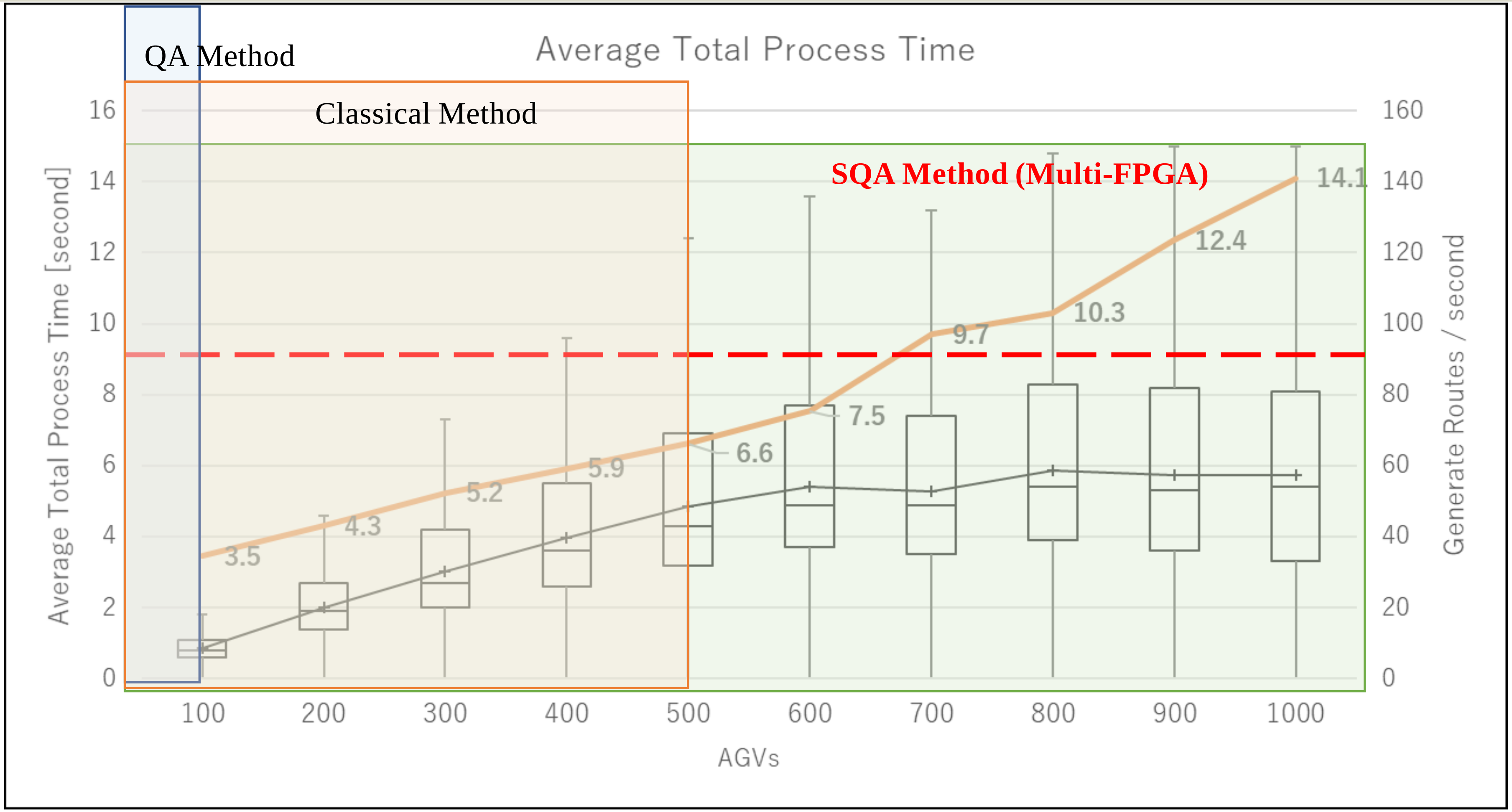
2. Related work: Basic of optimization methods

Be Original.

Comparison	Optimization Methods		
	Simulated Annealing (**SA**) [S. Kirkpatric et al., 1983]	Quantum Annealing (**QA**) [Kasowaki, and Nishimori, 1998]	Simulated Quantum Annealing (**SQA**) [E. Crosson, and A.W. Harrow, 1998]
Hardware Makers	Toshiba, Hitachi, Fujitu, NEC,…	D-wave	**SHARP**
Method overview	**Use thermal fluctuations in the annealing process to find the ground state:**  	**Use quantum fluctuations through a transverse magnetic field to find the ground state:**  	**Simulating the Quantum Annealing process on classical computer by** Path-integral Quantum Monte Carlo method: simulate the quantum tunneling phenomena of Ising model with transverse field. Use multiple replicas called "Trotters" to represent superposition of spins.  [M.W. Hashita, and M. Hariyama, 2018]
Solution quality	Depend on problem scale/complex	High	High
Industrial needs	Low	High	High (large scale)
Target problems	Simple & small scale	Qubits limited (5,000+)	Low speed

SQA with multi-FPGA is an innovative candidate to solve the large scale and complex problems

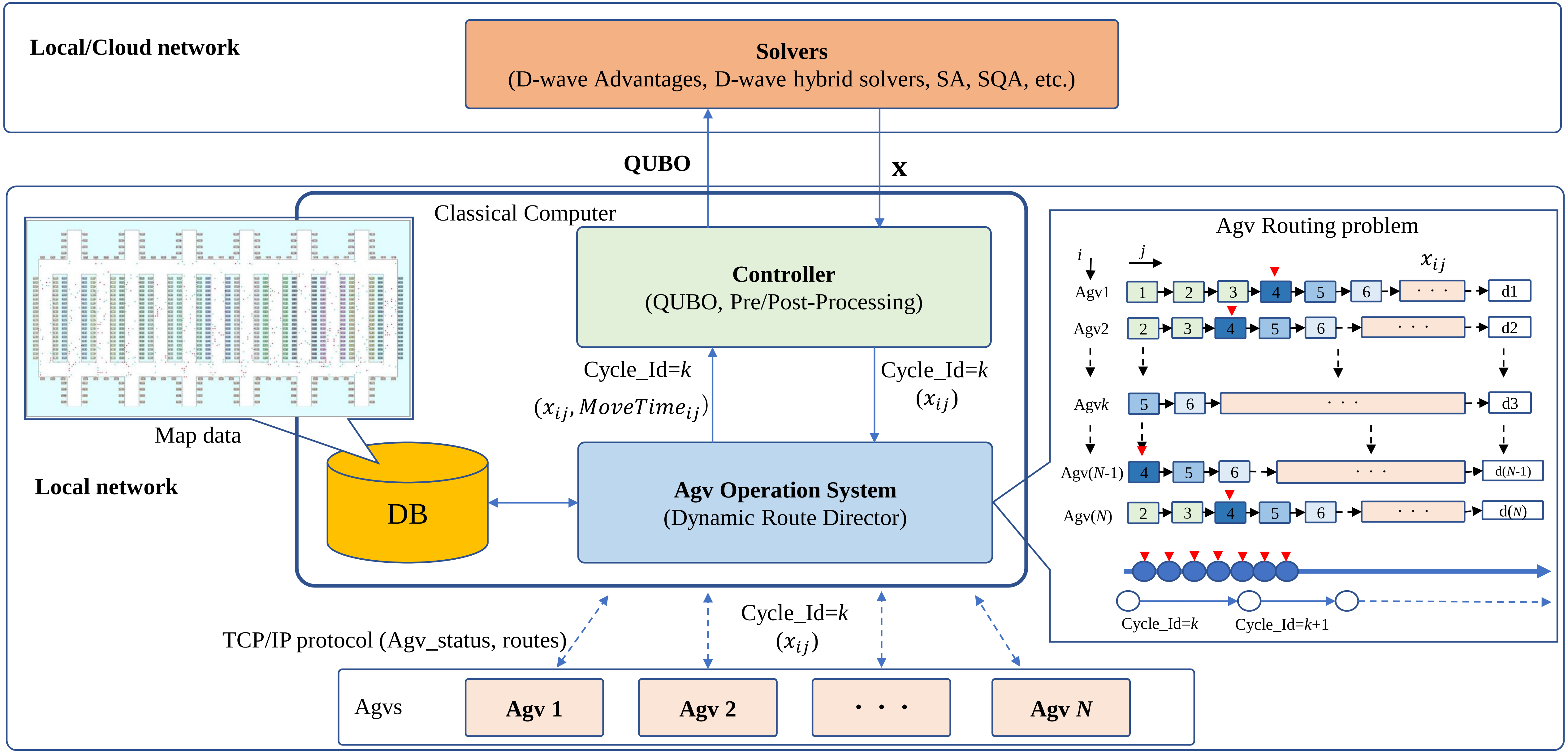
2. Related work: Motivation to use FPGAs to Speed-up the SQA method



Current AGV control system performances [N.Q Thinh, et.al., 2024]

We propose a Simulated Quantum Annealing accelerator using multi-FPGA for optimizing large-scale routing of 1000 AGVs

3. Problem formulation: Overview of real-world large-scale AGV Operation system



Local/Cloud network

Solvers
(D-wave Advantages, D-wave hybrid solvers, SA, SQA, etc.)

QUBO

X

Classical Computer

Controller
(QUBO, Pre/Post-Processing)

Cycle_Id=k
$(x_{ij}, MoveTime_{ij})$

Cycle_Id=k
(x_{ij})

Map data

DB

Agv Operation System
(Dynamic Route Director)

Local network

Agv Routing problem

i j x_{ij}

Agv1: 1 → 2 → 3 → 4 → 5 → 6 → · · · → d1

Agv2: 2 → 3 → 4 → 5 → 6 → · · · → d2

Agv$k$: 5 → 6 → · · · → d3

Agv(N-1): 4 → 5 → 6 → · · · → d(N-1)

Agv(N): 2 → 3 → 4 → 5 → 6 → · · · → d(N)

Cycle_Id=k Cycle_Id=k+1

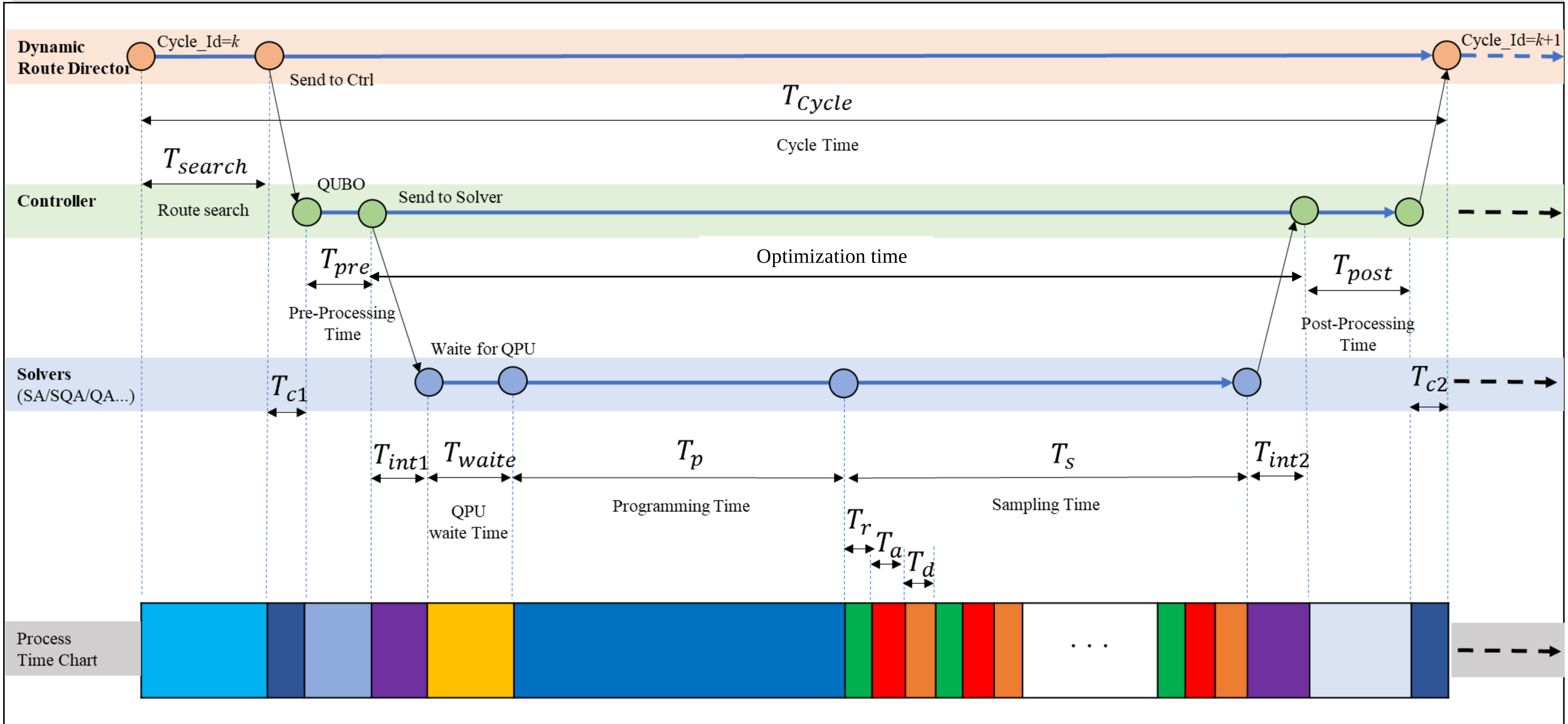
TCP/IP protocol (Agv_status, routes)

Cycle_Id=k
(x_{ij})

Agvs **Agv 1** **Agv 2** · · · **Agv N**

3. Problem formulation: Process flow

SHARP Be Original.



Route Optimization Process Flow

3. Problem formulation: Proposal object function

Object function:

Efficiency

$$H_1 = \sum_i \sum_j (\mathrm{MoveTimeAll}_{i,j} - \mathrm{MoveTimeNext}_{i,j}) \mathrm{X}_{i,j} \quad (2)$$

Constraint 1:

$$H_2 = \sum_i \left(\sum_j \mathrm{X}_{i,j} - 1 \right)^2 \quad (3)$$

Constraint 2:

$$H_3 = \sum_{\mathrm{tag}} \left(\sum_i \sum_j \mathrm{C}_{i,j,\mathrm{tag}} \mathrm{Pri}_i \mathrm{X}_{i,j} - 1 \right)^2 \quad (4)$$

Safety Control

QUBO formulation:

$$H = H_1 + \lambda_1 H_2 + \lambda_2 H_3 \quad (5)$$

Expected to improve Safety control and Efficiency

The objective function *H*1 using Equation (2) ,
Parameters:

- MoveTimeAll*i, j* : Move time to run from the current position to the destination position for AGV *i* using route *j* .
- MoveTimeNext*i, j* : Move time to execute the next optimized route from the current position for AGV *i* using route *j* .
- X*i, j* : QUBO variable indicating whether an AGV*i* uses route *j* . The value of this variable is either 0 or 1.

"MoveTimeAll*i, j*–MoveTimeNext*i, j* " represents the remaining move time of AGV *i* using route *j* .

Constraint *H*2 ensures that each AGV uses only a single route among all candidate routes.

Constraint *H*3 ensures that the collisions among AGV routes are avoided while prioritizing the routing of critical AGVs.

Parameters:

- C*i, j,tag* is a binary constant that has a value of 1 when AGV *i* using route *j* and that route goes over the position "*tag*". Otherwise, it takes the value 0.
- Pri*i* is the priority constant of AGV *i*. This value is determined by the priority of the tasks of AGVs such as going to the charging station, returning home, waiting, etc.

4. Proposed solution: Multi-FPGA SQA accelerator, system architecture, and Prototyping

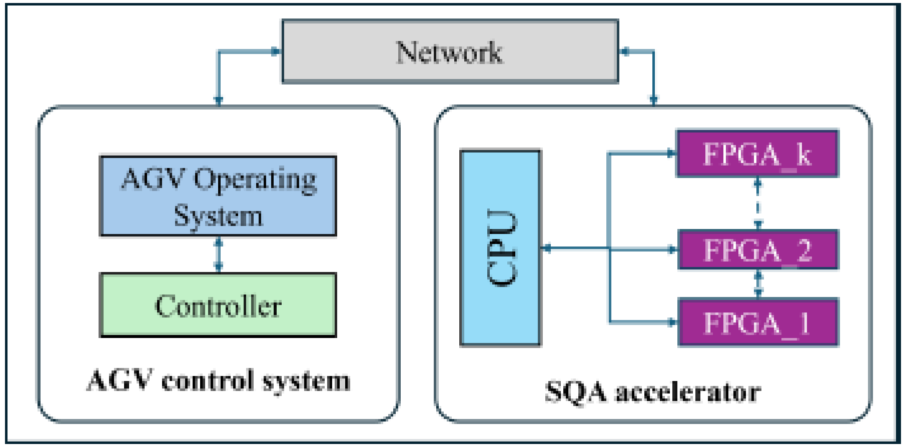


Figure 3: AGV control system architecture.

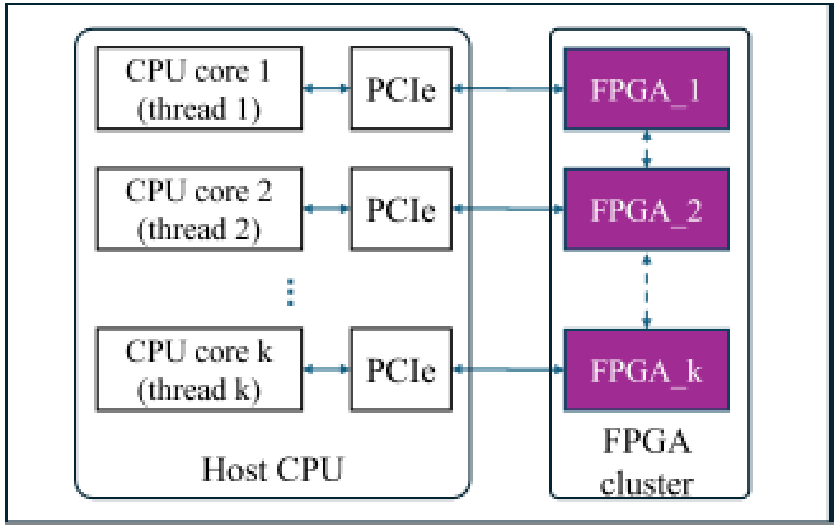
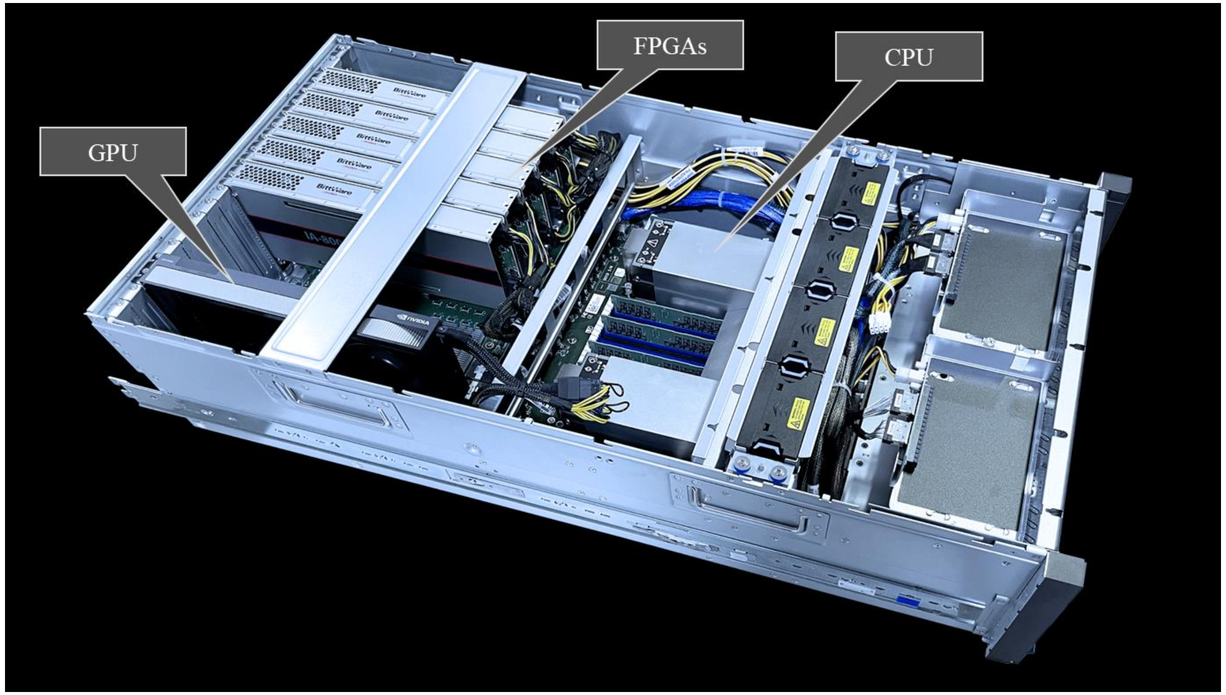


Figure 4: Multi-FPGA accelerator architecture.



The prototyping SQA accelerator

[N.Q Thinh, et.al., 2024]

4. Proposed solution: Multi-FPGA SQA accelerator architecture, and its control flow

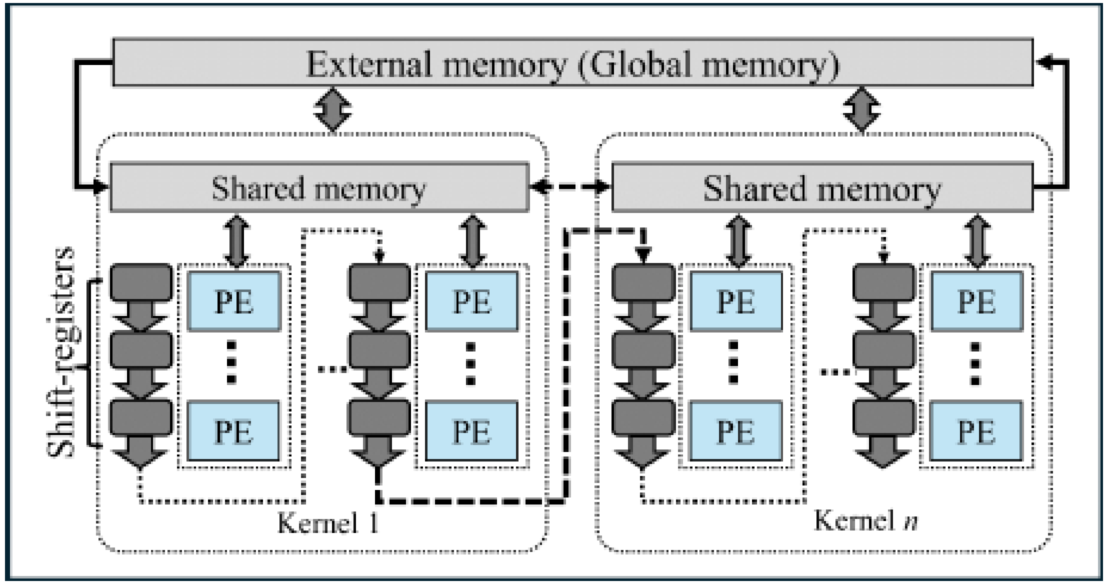


Figure 5: FPGA accelerator architecture.

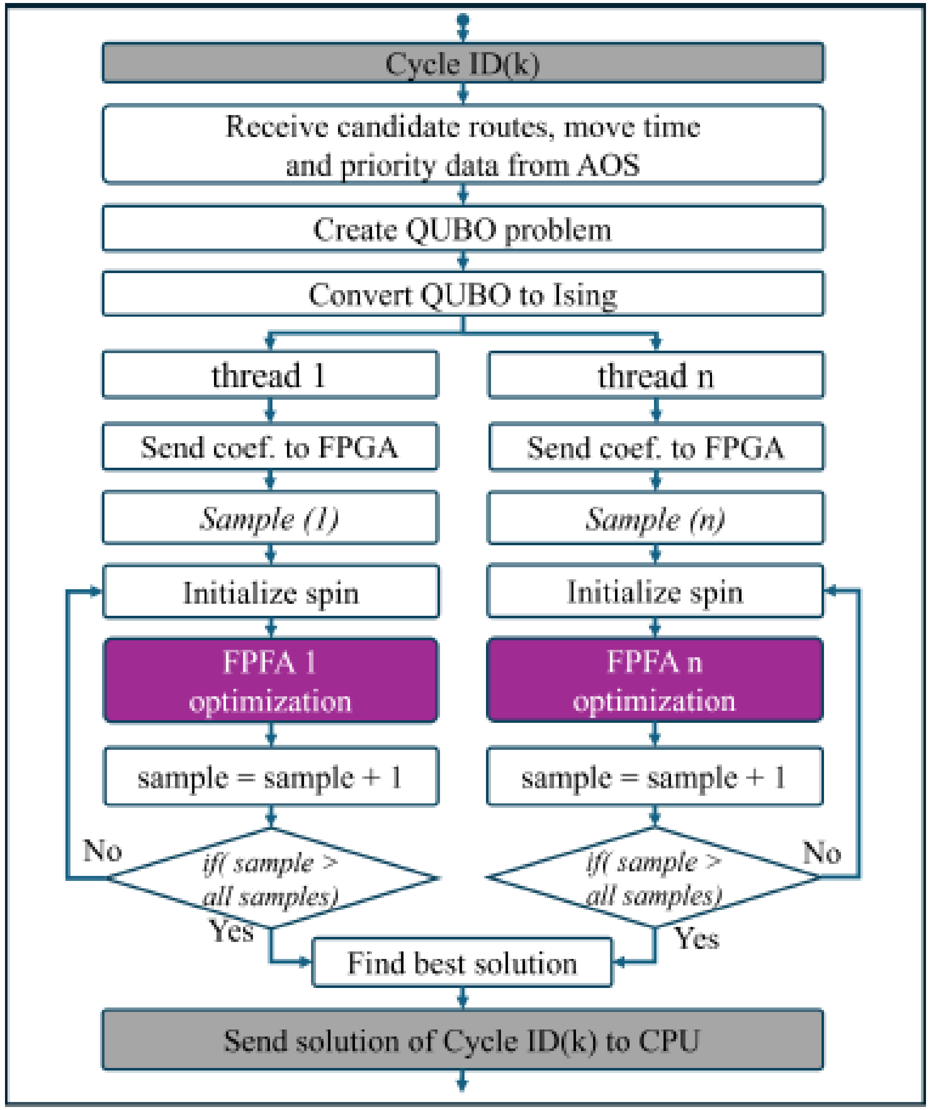


Figure 6: Control-flow of the multi-FPGA accelertor.

5. Evaluation: The specifications of the experimental

- **AGV control system:**

– Server: 24-core 3.2GHz Intel core i9-14900KF CPU, 32GB DDR4 memory, 1.18TB SSD.

– AGV Simulator

- **OpenJij Accelerator:**

– PC: 12-core Apple M2 pro CPU, 32GB memory, 2TB SSD.

– Compiler: python 3.9.13 with OpenJij and D-wave dimod.

- **SQA Accelerator:**

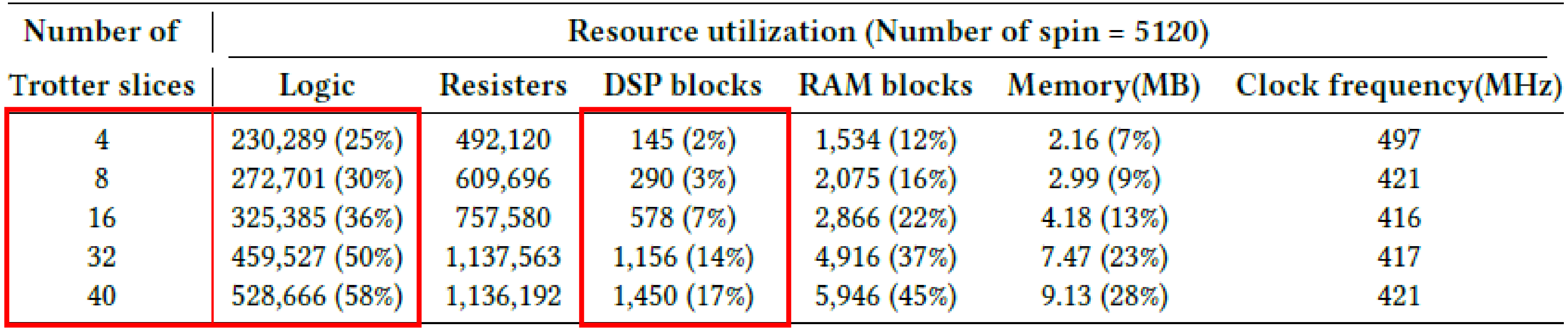
– Server: Dual Intel Xeon Gold 6326 (16C/32T, 2.9 GHz) CPU, 256GB 8-channel DDR4 memory, 240GB SSD.

– FPGA: 5 Intel Agilex 7 IA840F FPGA boards [1].

– Compiler: Quatus 21.4 compiler with Intel FPGA SDK for OpenCL and Intel C compiler version 2023.1.

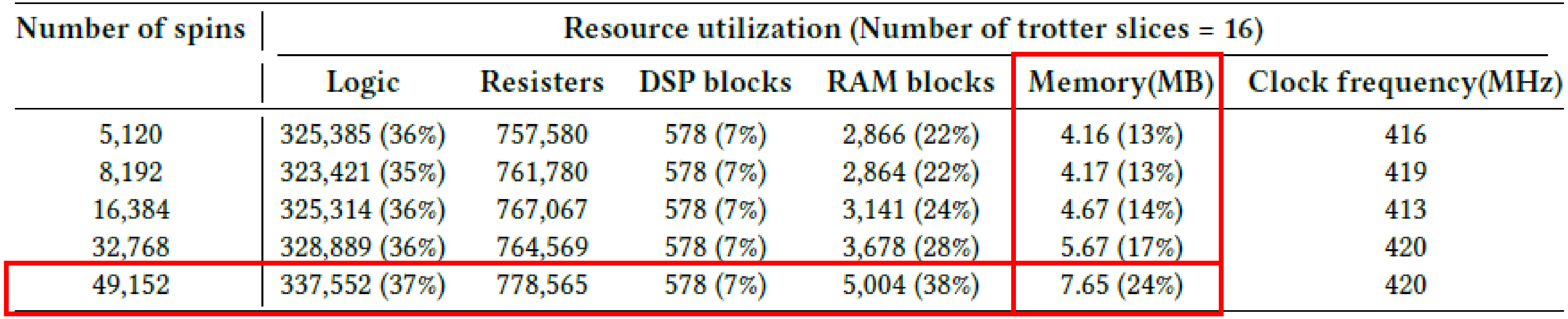
5. Evaluation: Comparison of the FPGA resource utilization

Table 1: Comparison of the FPGA resource utilization for different numbers of Trotter slices



Number of	Resource utilization (Number of spin = 5120)					
Trotter slices	Logic	Resisters	DSP blocks	RAM blocks	Memory(MB)	Clock frequency(MHz)
4	230,289 (25%)	492,120	145 (2%)	1,534 (12%)	2.16 (7%)	497
8	272,701 (30%)	609,696	290 (3%)	2,075 (16%)	2.99 (9%)	421
16	325,385 (36%)	757,580	578 (7%)	2,866 (22%)	4.18 (13%)	416
32	459,527 (50%)	1,137,563	1,156 (14%)	4,916 (37%)	7.47 (23%)	417
40	528,666 (58%)	1,136,192	1,450 (17%)	5,946 (45%)	9.13 (28%)	421

Table 2: Comparison of the FPGA resource utilization for different numbers of spins



Number of spins	Resource utilization (Number of trotter slices = 16)					
	Logic	Resisters	DSP blocks	RAM blocks	Memory(MB)	Clock frequency(MHz)
5,120	325,385 (36%)	757,580	578 (7%)	2,866 (22%)	4.16 (13%)	416
8,192	323,421 (35%)	761,780	578 (7%)	2,864 (22%)	4.17 (13%)	419
16,384	325,314 (36%)	767,067	578 (7%)	3,141 (24%)	4.67 (14%)	413
32,768	328,889 (36%)	764,569	578 (7%)	3,678 (28%)	5.67 (17%)	420
49,152	337,552 (37%)	778,565	578 (7%)	5,004 (38%)	7.65 (24%)	420

Discussion:

- Using more Trotter slices usually provides better results in less time consuming more resources.
- The logic resources is the most critical factor while the DSP usage is very small.
- 49,152 spins without utilizing less than 40% of the FPGA resources.
- We can process more spins using an FPGA with a larger externel memory

5. Evaluation: Comparison of the processing time of multi-FPGAs

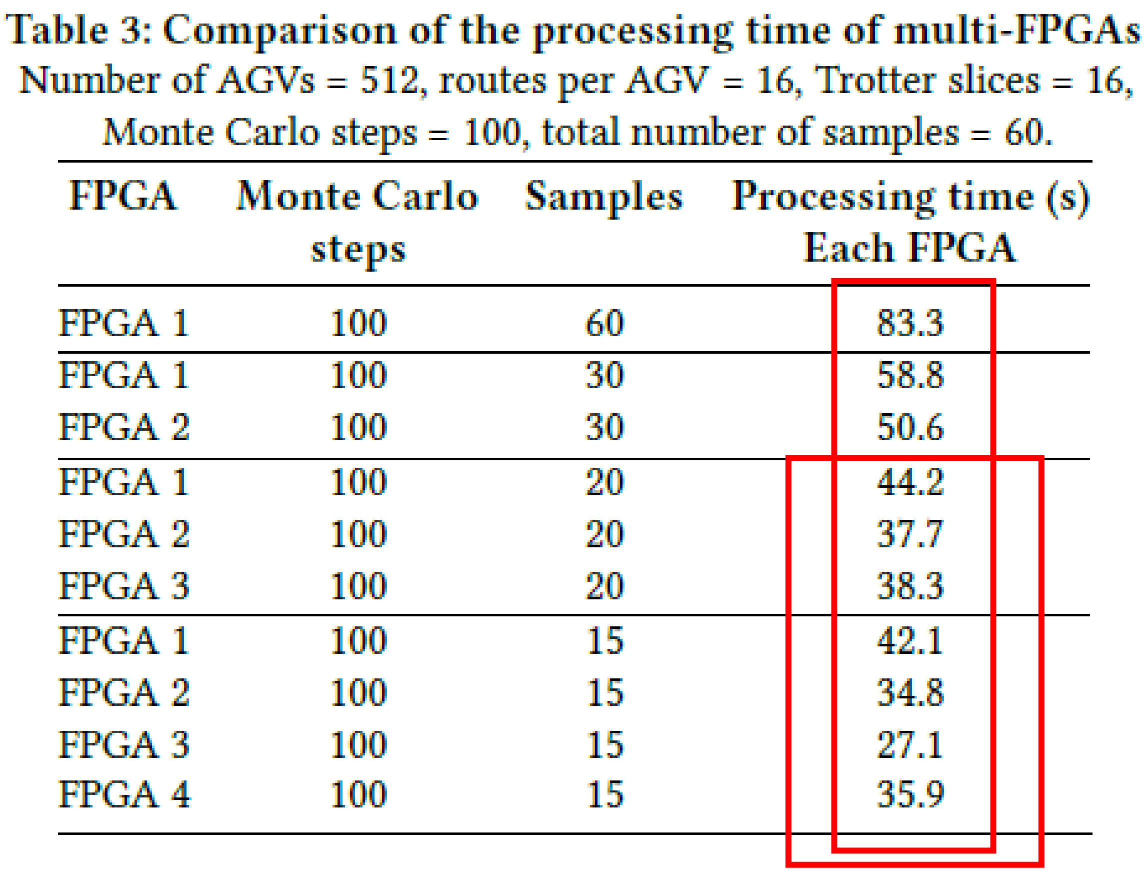


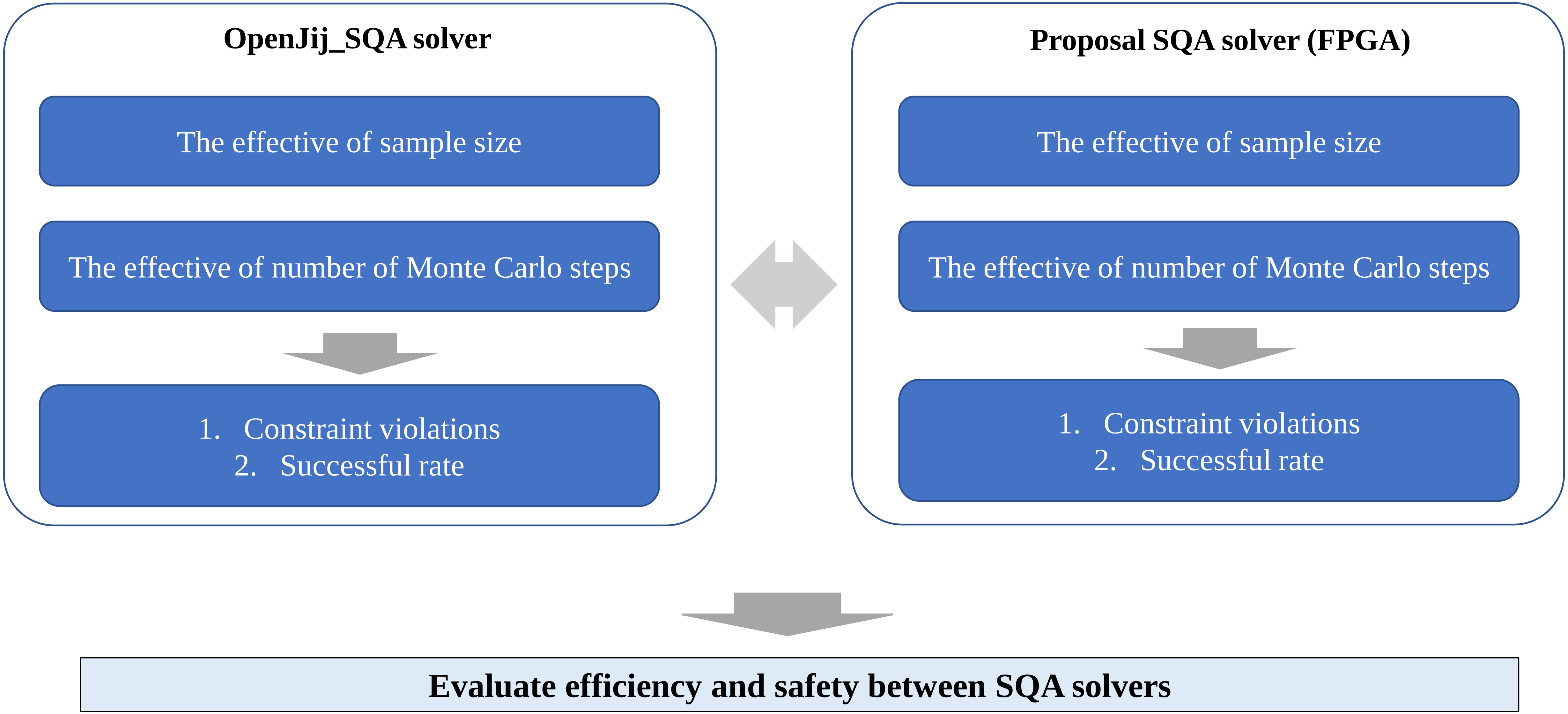
Table 3: Comparison of the processing time of multi-FPGAs
Number of AGVs = 512, routes per AGV = 16, Trotter slices = 16,
Monte Carlo steps = 100, total number of samples = 60.

FPGA	Monte Carlo steps	Samples	Processing time (s) Each FPGA
FPGA 1	100	60	83.3
FPGA 1	100	30	58.8
FPGA 2	100	30	50.6
FPGA 1	100	20	44.2
FPGA 2	100	20	37.7
FPGA 3	100	20	38.3
FPGA 1	100	15	42.1
FPGA 2	100	15	34.8
FPGA 3	100	15	27.1
FPGA 4	100	15	35.9

Discussion:

- The number of samples per FPGA decreases when using more FPGAs. The total processing time also decreases.
- The process times of other experiments almost the same. The reason for this difference could be the synchronization overhead by the CPU in multi-thread implementation.

5. Evaluation: Comparison method of SQA solvers(1/2)



OpenJij_SQA solver

The effective of sample size

The effective of number of Monte Carlo steps

1. Constraint violations
2. Successful rate

Proposal SQA solver (FPGA)

The effective of sample size

The effective of number of Monte Carlo steps

1. Constraint violations
2. Successful rate

Evaluate efficiency and safety between SQA solvers

5. Evaluation: Comparison method of SQA solvers

In order to evaluate the quality, we measure the following constraint violations and routing availability.

- No routes : No routes are available due to the violation of the “single route per AGV” constraint given by Equation 3
- Multi routes : Multiple routes are found due to the violation of “single route per AGV” constraint given by Equation 3
- Collisions : Routes collide with each other violating the “collision avoidance” constraint given by Equation 4.
- No movement : The route found does not cause any movement for the AGV.
- Successful routes : Successful AGV movement to a new destination tag safely without colliding with others.

$$H_2 = \sum_i \left(\sum_j X_{i,j} - 1 \right)^2 \tag{3}$$

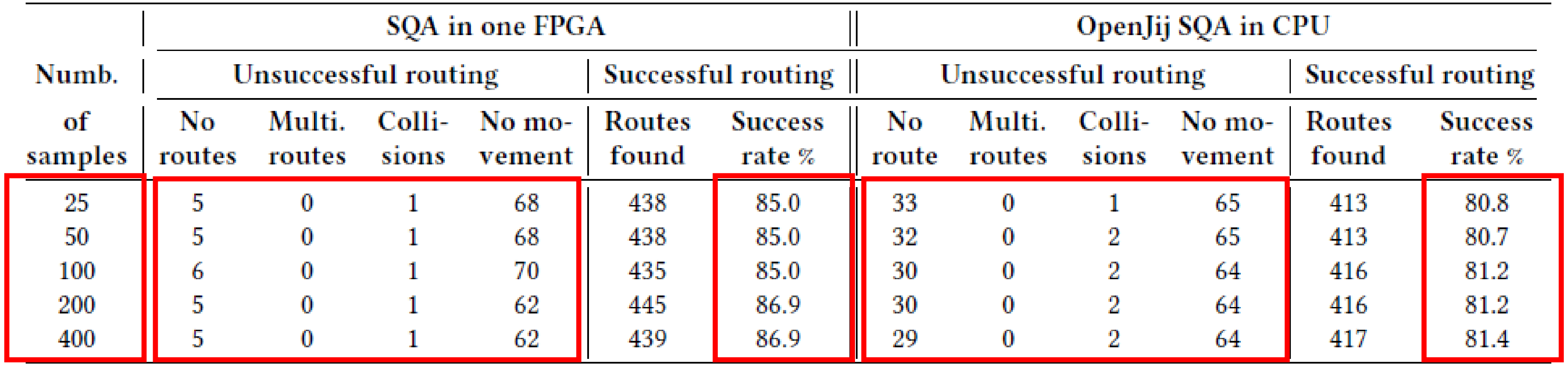
$$H_3 = \sum_{\text{tag}} \left(\sum_i \sum_j C_{i,j,\text{tag}} \text{Pri}_i X_{i,j} - 1 \right)^2 \tag{4}$$

The success rate of the routing is calculated by Equation (6).

$$\text{Success rate} = \frac{\text{Successul routes}}{\text{All routes}} \times 100\% \tag{6}$$

5. Evaluation: Comparison results between SQAs(1/2)

Table 4: Evaluation of the effect of the sample size.
Number of AGVs = 512, routes per AGV = 16, number of Trotter slices = 16, number of Monte Carlo steps = 100.



	SQA in one FPGA						OpenJij SQA in CPU					
Numb.	Unsuccessful routing				Successful routing		Unsuccessful routing				Successful routing	
of samples	No routes	Multi. routes	Collisions	No movement	Routes found	Success rate %	No route	Multi. routes	Collisions	No movement	Routes found	Success rate %
25	5	0	1	68	438	85.0	33	0	1	65	413	80.8
50	5	0	1	68	438	85.0	32	0	2	65	413	80.7
100	6	0	1	70	435	85.0	30	0	2	64	416	81.2
200	5	0	1	62	445	86.9	30	0	2	64	416	81.2
400	5	0	1	62	439	86.9	29	0	2	64	417	81.4

Discussion: The effect of the sample size for both methods

- Same parameters conditions: Number of samples, Trotter slices, Monte Carlo steps, AGVs, Routes per AGV….
- The FPGA accelerator has a better success rate, and small number of routing violations compared to OpenJij.
- Changing sample size does not show any difference, possibly due to the sufficiently large sample size.

5. Evaluation: Comparison results between SQAs(2/2)

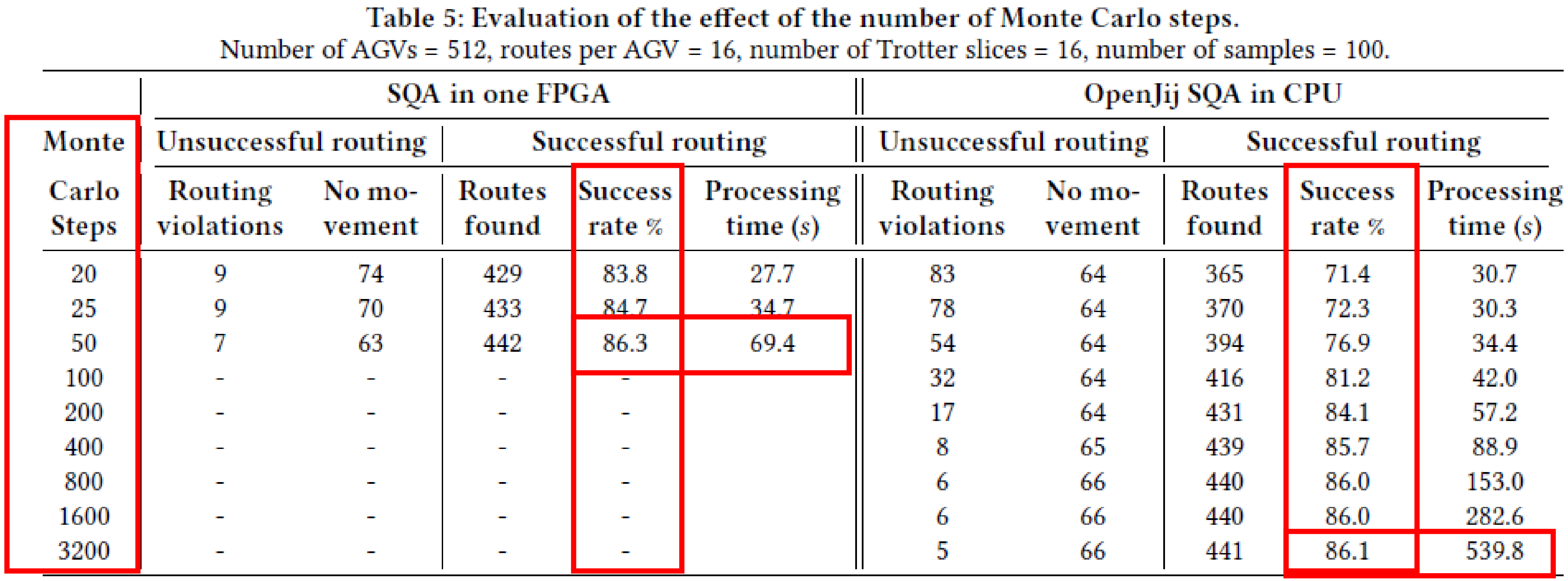


Table 5: Evaluation of the effect of the number of Monte Carlo steps.
Number of AGVs = 512, routes per AGV = 16, number of Trotter slices = 16, number of samples = 100.

	SQA in one FPGA					OpenJij SQA in CPU				
Monte	Unsuccessful routing		Successful routing			Unsuccessful routing		Successful routing		
Carlo Steps	Routing violations	No movement	Routes found	Success rate %	Processing time (s)	Routing violations	No movement	Routes found	Success rate %	Processing time (s)
20	9	74	429	83.8	27.7	83	64	365	71.4	30.7
25	9	70	433	84.7	34.7	78	64	370	72.3	30.3
50	7	63	442	86.3	69.4	54	64	394	76.9	34.4
100	-	-	-	-		32	64	416	81.2	42.0
200	-	-	-	-		17	64	431	84.1	57.2
400	-	-	-	-		8	65	439	85.7	88.9
800	-	-	-	-		6	66	440	86.0	153.0
1600	-	-	-	-		6	66	440	86.0	282.6
3200	-	-	-	-		5	66	441	86.1	539.8

Discussion: The effect of the number of Monte Carl steps for both methods

- The success rate is increased with the number of Monte Carlo steps.
- The FPGA accelerator achieved a success rate of 86.3% in just 50 Monte Carlo step.
- This success rate is not achieved by the OpenJij SQA in CPU, even using 3200 Monte Carlo steps.
- Compare the processing times when the success rate of 86.3%, the proposed accelerator is nearly 8 times faster than OpenJij

Note: Additional experiments confirmed that the processing time increases exponentially with the increase in the number of Monte Carlo steps."

6. Conclusion

(1) We proposed a novel QUBO formulation for real-world large-scale AGV routing optimization.

(2) We introduced a scalable multi-FPGA SQA accelerator for large-scale problems with thousands of AGVs. Each FPGA processes up to 50,000 variables, far exceeding the capacity of the D-Wave Advantage machine. The SQA accelerator demonstrated faster processing speeds, fewer routing violations, and a much higher success rate compared to the existing OpenJij SQA solver on CPU.

(3) We fully integrated the SQA solver into the AGV control system and observed correct behavior for large-scale AGV routing.

(4) We also identified Challenges and future directions include:

- Underutilization of FPGA resources and restricted processing speed due to small external memory bandwidth, which can be solved with high-bandwidth memory in future FPGAs.
- Thread synchronization overhead causing processing time variations across FPGAs, requiring further investigation.
- Future system-wide evaluations are needed to assess the overall advantages of the multi-FPGA setup.

6. Conclusion: Demonstration with real SQA Accelerators and real AGVs

Thank you for your attention!