



FlowReID - An integer programming approach to re-identification in livestock tracking using quantum and classical optimization

L. Scavuzzo^a, J.C. Boschero^a, J.E. Doornweerd^b, J. Vandenplas^b, S.G.G. Cramer^a, E.P. Aguilera^a, T. Pook^{b,*}

^a TNO, Applied Cryptography and Quantum Algorithms, 2595 DA Den Haag, the Netherlands

^b Wageningen University & Research, Animal Breeding and Genomics Group, 6708 PB Wageningen, the Netherlands

ARTICLE INFO

Keywords:

Quantum computing
Computer vision
Multi-object tracking
Re-identification

ABSTRACT

Accurate tracking of individual animals is essential for precision livestock farming, enabling health monitoring, behavior analysis, and phenotyping. Multi-object tracking in livestock environments is challenging due to frequent occlusions, close interactions, and visually similar animals. In this study, we present FlowReID, a targeted post-processing re-identification (reID) approach that extends the BoT-SORT tracking framework to make it suitable for optimization using quantum computing. For this, we formulate the reID problem as a binary linear program and subsequently reformulate it as a Quadratic Unconstrained Binary Optimization (QUBO), enabling optimization using a classical solver, simulated annealing, or quantum annealing.

Evaluation on video sequences of 16 Holstein Friesian cows from eight camera views demonstrated that FlowReID consistently improved identity consistency compared to the standard Ultralytics reID module, reducing the number of identity switches from 9 to 5, while improving overall multi-object tracking metrics like HOTA (89.7 to 90.8), MOTA (95.4 to 96.3), and IDF1 (96.0 to 97.4).

Although quantum solvers were evaluated, they did not outperform classical methods in either speed or accuracy under current quantum hardware constraints. While the proposed algorithm is quantum-ready from a formulation perspective, we did not observe a quantum advantage. Nevertheless, this work provides a proof-of-concept for implementing multi-object tracking in precision livestock farming in a quantum-compatible optimization framework. This formulation enables direct transfer to future quantum or hybrid solvers as quantum hardware continues to mature. Since FlowReID is also implemented using classical optimization solvers, our newly developed approach can be deployed without requiring access to quantum computing hardware.

1. Introduction

Multi-object tracking (MOT) is important for modern precision livestock farming as it enables simultaneous tracking of multiple animals in complex livestock environments [1]. The resulting video tracks or tracklets of the animals can later be used for animal management as well as a phenotyping tool in animal breeding programs, for instance, to continuously assess the health and welfare of the animals. In general, multi-object tracking is a computer vision task that aims to detect and track all objects of interest in a scene, be it people, plants, or animals, with a consistent, unique identifier.

In precision livestock farming, multi-object tracking is often approached with tracking-by-detection [1]. Tracking-by-detection uses two algorithms, one to detect the object of interest, such as You Only Look Once (YOLO, [2]), and one to track the detected objects of inter-

est, such as Simple Online and Realtime Tracking (SORT, [3]) or its enhanced variant BoT-SORT [4], which incorporates camera motion modeling and appearance features for improved identity preservation. However, assigning a consistent identifier to each video track is difficult due to occlusions and frequent interactions between animals. As a result, detections can be lost, and tracks may be re-initialized, or identities may be switched between animals and frames. To mitigate this issue, trackers tend to include a re-identification (reID) approach to recover lost or switched identities to maintain a consistent identity for each video track [5,6]. ReID could use, for instance, appearance and geometric features to recover identities [7,8]. For example, DeepSORT integrated appearance features in SORT, which reduced identity switches by 45% on the MOT16 benchmark (pedestrians) [9]. However, in groups of similar-looking animals, appearance features contribute less to identity recovery, and reID becomes a re-association problem.

* Corresponding author.

E-mail address: torsten.pook@wur.nl (T. Pook).

<https://doi.org/10.1016/j.atech.2026.102498>

Received 18 March 2026; Received in revised form 12 August 2026; Accepted 19 August 2026

Available online 20 August 2026

2772-3755/© 2026 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Today, most computer vision tasks are executed on GPU hardware [10]; however, the identity-association of multi-object tracking reduces to a combinatorial optimization problem. As the number of tracked animals increases, these problems can grow exponentially in complexity, motivating the exploration of quantum algorithms that may offer computational advantages. Quantum computing is an emerging technology offering a new approach to computation, and its potential applications in agriculture have only recently begun to be explored [11,12]. In contrast to bits on traditional computing hardware (CPUs and GPUs), quantum computers operate on quantum bits (qubits). Qubits reflect pieces of quantum information that may exhibit two intrinsic quantum properties: superposition, allowing them to exist in multiple states simultaneously, and entanglement, enabling links between qubits [13]. This unique storage structure allows quantum systems to process information in ways that have no classical counterpart and enables the design of specialized quantum algorithms. As such, a quantum computer is not intrinsically faster at performing the same operations, but can potentially enable the use of new algorithms that solve the same computational problems faster to achieve a ‘quantum advantage’ [14]. A well-known example of quantum advantage is Shor’s algorithm [15] to calculate prime factors of large integers exponentially faster than the best-known classical methods, potentially rendering current encryption schemes vulnerable once scalable quantum hardware becomes available. Although such hardware is not yet available, major technology companies [16,17] and research organizations [18,19] are heavily investing in the development of large-scale fault-tolerant quantum computers capable of executing these previously developed quantum algorithms in practice.

In the context of combinatorial optimization, quantum computing offers several frameworks. While gate-based quantum algorithms such as the Quantum Approximate Optimization Algorithm (QAOA, [20]), Quantum Semi-definite Programming [21], and Quantum Least Squares Fitting [22] have been proposed for such tasks, quantum annealers are particularly well suited for optimization problems that can be expressed as Quadratic Unconstrained Binary Optimization (QUBO) models. QUBO-based formulations have previously been investigated for quantum-enhanced multi-object tracking by [23] and [24]. Although the application is similar, [23] proposed the first multi-object tracking formulation tailored for adiabatic quantum computing that replaces traditional network-flow models with assignment matrices, allowing a non-constrained quadratic binary optimization representation that scales linearly with detections, tracks, and frames. Constraints are incorporated via Lagrangian multipliers, and the formulation is optimized for Ising-based quantum annealers. However, this approach requires replacing the traditional workflow end-to-end with a quantum-based solution, which also includes replacing parts of the pipeline where classical approaches already excel. Ihara [24] focuses on enhancing multi-object tracking accuracy and speed in traffic scenarios using quantum annealing and reverse annealing; their approach also requires replacing the conventional tracking pipeline with a quantum-centric workflow. Furthermore, [24] emphasizes ensemble processing and annealing schedule optimization for real-time urban traffic applications, whereas our work prioritizes scalability and robustness in crowded agricultural environments with frequent occlusions.

This work evaluates the feasibility of quantum computing in precision livestock farming through a proof-of-concept integration into the BoT-SORT tracking pipeline. The primary novelty lies not in the use of QUBO itself, but in the introduction of a targeted post-processing reID framework that preserves high-performance tracking while enabling seamless integration with both classical and quantum optimization methods. Specifically, the framework performs pairwise comparisons of tracked objects across consecutive time steps based on bounding box overlap and reformulates the reID problem as a binary integer program that can be solved using either classical optimization techniques or quantum hardware.

The remainder of this paper is structured as follows. Section 2 introduces the methodological framework, including the binary linear pro-

gram (BLP) and QUBO formulations as well as the dataset, experimental setup, and solver configurations. Section 3 presents the benchmarking results for both re-identification performance and solver comparisons. Finally, Section 4 discusses the implications and future directions for scalable quantum-enhanced tracking.

2. Material & methods

2.1. Multi-object tracking problem

The novelty of this work lies in its formulation of the tracking problem using integer programming, in particular, a binary linear program. The input to a tracking framework is a video, i.e., a sequence of frames. At its core, tracking can be understood as the task of matching detections across multiple frames. Abstractly, this problem can be represented as a layered network, where each layer represents a frame, and each node in a layer represents a detection. We draw edges between any two nodes in consecutive layers. Edges have an associated weight that represents the cost of associating detections across frames, which is described in more detail in Section 2.1.2. Note that the number of nodes per layer is not necessarily constant. A path from the initial layer to the last is a possible trajectory of a tracked object. Finally, we can find a minimum cost tracking by optimizing the cost of a set of non-intersecting paths.

Fig. 1(a) illustrates an example visualization of such a graph structure. In this scenario, three detections occur at the initial time step, while four detections are observed at some of the intermediate steps, as well as the final time step. This could be due to false (or double) detections, or it may represent cases where an object enters and exits the frame. Fig. 1(b) demonstrates a feasible solution to this tracking problem, where the three objects detected in the first frame are matched throughout the frame sequence to obtain consistent trajectories.

In the following subsections, we present the mathematical formulation that models this graph problem. In particular, this is a case of a minimum cost multi-commodity network flow problem with edge capacities [25]. Subsequently, we discuss the chosen cost function and the overall pipeline of our framework.

2.1.1. Binary linear programming formulation

Let T be the number of frames in the considered sequence, hereafter referred to as the number of time steps. Frames are indexed by $t \in \{0, \dots, T-1\}$. Each frame t has an associated layer in the graph with d_t nodes, each representing a detection. For $t \in \{0, \dots, T-2\}$, $i \in \{1, \dots, d_t\}$ and $j \in \{1, \dots, d_{t+1}\}$ we define the binary decision variables x_{ijt} , which represent whether detection i in time step t and detection j in time step $t+1$ correspond to the same individual. Equivalently, for a valid tuple (i, j, t) , let c_{ijt} be the cost of associating detection i in time step t to detection j in time step $t+1$ as per some later introduced cost metric. The tracking problem can then be defined as:

$$\text{Minimize} \quad \sum_{t=0}^{T-2} \sum_{i=1}^{d_t} \sum_{j=1}^{d_{t+1}} c_{ijt} x_{ijt} \quad (1)$$

$$\text{Subject to} \quad \sum_{j=1}^{d_{t+1}} x_{ijt} \leq 1, \quad i = 1, \dots, d_t, \quad t = 1, \dots, T-2 \quad (2)$$

$$\sum_{j=1}^{d_{t-1}} x_{ji(t-1)} = \sum_{j=1}^{d_{t+1}} x_{ijt}, \quad i = 1, \dots, d_t, \quad t = 1, \dots, T-2 \quad (3)$$

$$\sum_{j=1}^{d_1} x_{ij0} = 1, \quad i = 1, \dots, d_0 \quad (4)$$

$$x_{ijt} \in \{0, 1\} \quad (5)$$

- This model can be interpreted as follows. The objective is to optimize over the binary decision matrix $\mathbf{x}$, given the cost matrix $\mathbf{c}$. We have

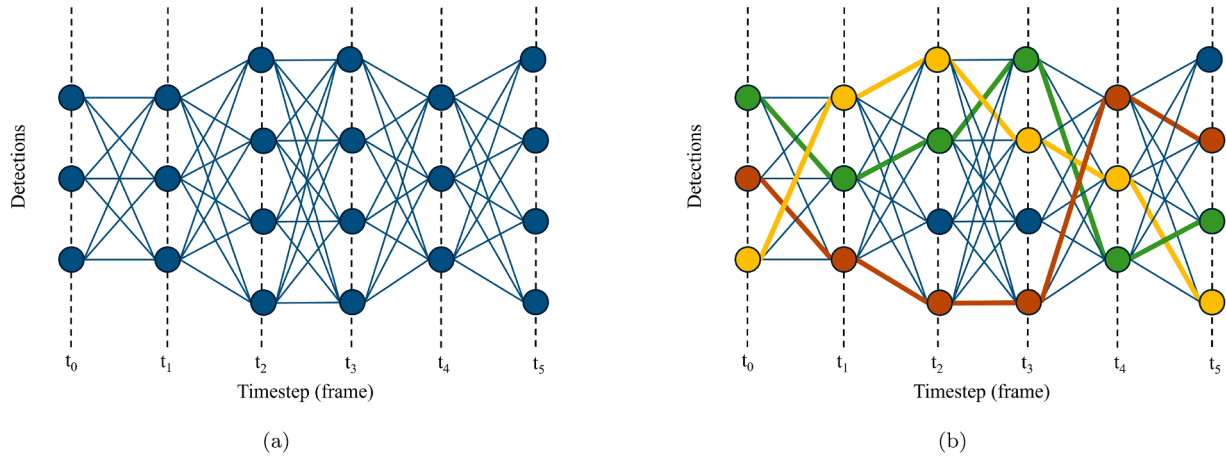


Fig. 1. Example network formulation of an object tracking problem of six subsequent time steps (a) and an exemplary solution of the tracking problem (b) with each tracklet indicated by a different color and blue nodes indicating detections that were not assigned to a tracklet. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

three types of constraints. The first constraint (2) ensures that each detection can be associated with at most one object by ensuring that for each node only one outgoing edge is chosen. The second constraint (3) enforces consistency in the number of connections between consecutive time steps. This is done by imposing that, for each node, the same number of incoming and outgoing edges are selected (either one or none). This guarantees temporal continuity. The third constraint (4) requires that all detections in the initial frame of the video sequence must be matched.

This model can also be interpreted as a network flow model [25], where units of flow are routed through the network from the initial to the last layer. In this context, the first constraint ensures a capacity constraint on the edges: at most one unit of flow can be routed on each edge. The second constraint ensures flow continuity: flow cannot be created in a node (except on the first layer); therefore, what comes in is what comes out. The last constraint is a boundary condition saying that one unit of flow is pumped out of each of the initial nodes.

There are two limitations to the model, which are that the initial frame of the video sequence must contain the true number of objects and that if the number of detections (i.e., nodes) is strictly less than the number of objects in the initial frame, no feasible solution exists. If this is the case, we can extend the model by adding edges in the following way. Let f_1 be the frame strictly before the frame where the number of detections goes below the initial one. Let $f_2 > f_1$ be the first frame where the number of detections is greater than or equal to the initial one. We draw edges connecting all nodes in f_1 to all nodes in f_2 . These edges are represented by binary decision variables y_{ij} with $i = 1, \dots, d_{f_1}$ and $j = 1, \dots, d_{f_2}$. We must then modify the constraints (2)–(4) associated with $t = f_1$ and $t = f_2$ in the following way:

$$\sum_{j=1}^{d_{f_1+1}} x_{ijf_1} + \sum_{j=1}^{d_{f_2}} y_{ij} \leq 1 \quad \text{for } i = 1, \dots, d_{f_1} \quad (6)$$

$$\sum_{j=1}^{d_{f_1-1}} x_{jif_1-1} = \sum_{j=1}^{d_{f_1+1}} x_{ijf_1} + \sum_{j=1}^{d_{f_2}} y_{ij} \quad \text{for } i = 1, \dots, d_{f_1} \quad (7)$$

$$\sum_{j=1}^{d_{f_2-1}} x_{jif_2-1} + \sum_{j=1}^{d_{f_1}} y_{ji} = \sum_{j=1}^{d_{f_2+1}} x_{ijf_2} \quad \text{for } i = 1, \dots, d_{f_2}. \quad (8)$$

As the goal of this work was primarily a proof-of-concept, we limit ourselves to the model in Eq. (1)–(5). The general formulation of this problem is polynomially solvable (proof can be found in Berclaz et al. [26]) because the underlying graph is layered and therefore the constraint matrix is totally unimodular. The model extension described in (6)–(8) makes the problem NP-hard. While an extension of the model

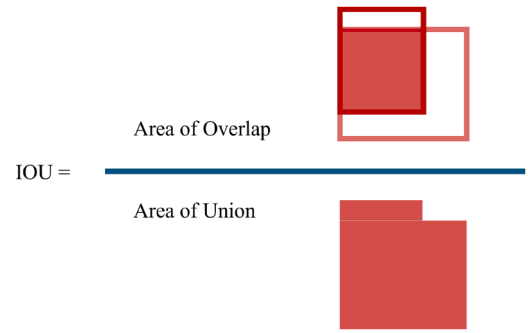


Fig. 2. A graphical representation of the intersection over union (IoU) between two bounding boxes. As the IoU value approaches one, the likelihood increases that the bounding boxes correspond to the same object in consecutive frames.

on traditional computing hardware would have been relatively straightforward, the transition to a quantum computational formulation would have required a substantially more complex reformulation of the problem structure, which was therefore not considered within the scope of this work.

2.1.2. Cost and physical representation

In livestock production, appearance is often not very distinct between animals. Different livestock breeds and species can have little visual variation between individuals, for instance, broiler chickens or Jersey cattle. Hence, we use the Intersection over Union (IoU) as the primary cost metric. The IoU is computed by dividing the area of overlap between two bounding boxes by the area of their union. By definition, IoU can only take values in $[0, 1]$. This concept is illustrated in Fig. 2. We want to maximize the IoU between matched detections. Since our problem is a cost minimization one, we define the cost to be $1 - \text{IoU}$.

To incorporate the cost metric into the model, the IoU must be computed for each combination of detections in two consecutive time steps. Intuitively, this results in a highly dense graph structure, where each node in one layer is connected to every node in the next, reflecting all possible associations between detections over time. In order to reduce the model size, one can remove the edges with associated $\text{IoU} = 0$ (equivalently, edges of cost 1).

An example of this process using cows is illustrated in Fig. 3. The initial frame shown in Fig. 3(a) contains the detections and the corresponding bounding box for each cow. Fig. 3(b) presents the same concept applied to the subsequent frame, t_2 . The next step involves comparing

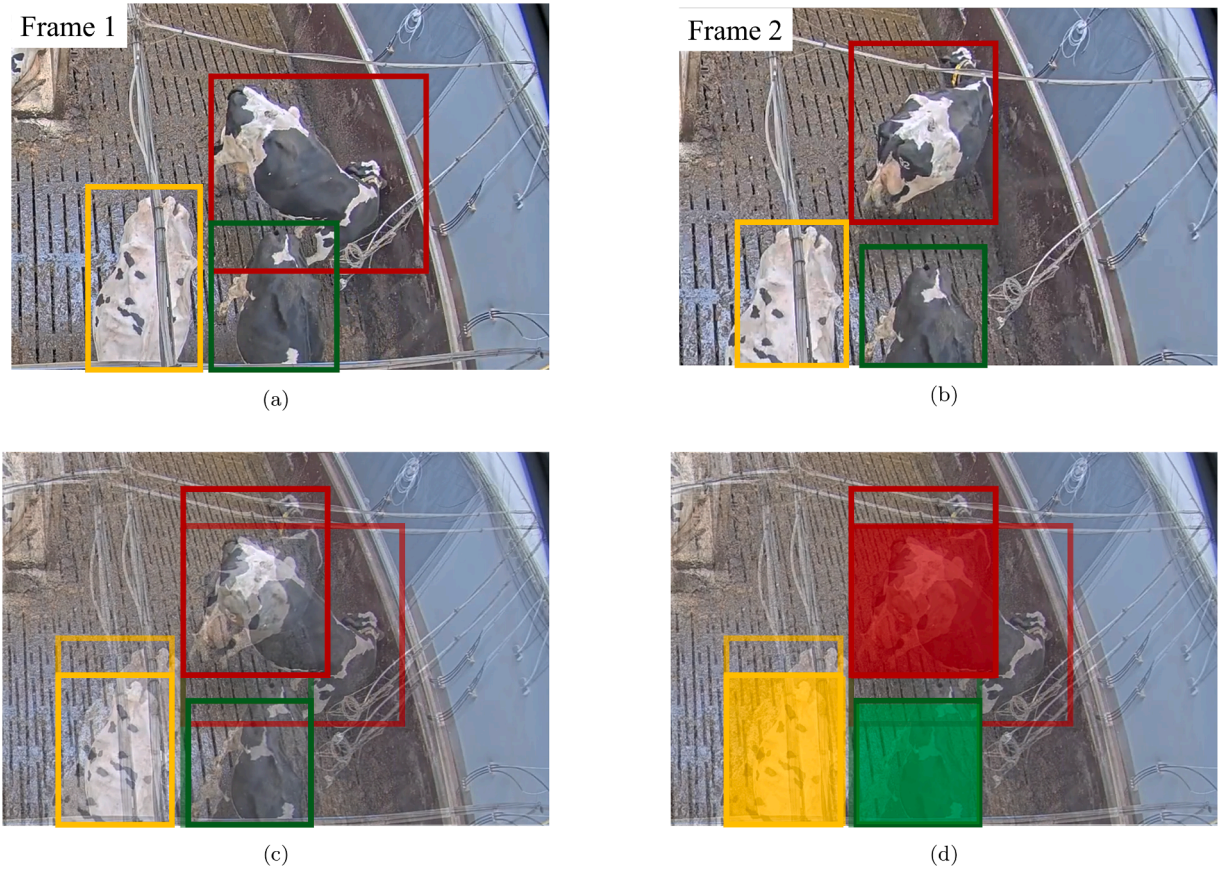


Fig. 3. Exemplary visualization of steps of the IoU calculation, including bounding boxes in the first (a) and second (b) frame, their overlay (c), and the intersection of union (d), with the bounding boxes of each cow indicated by a unique color. For simplicity, only the IoUs of the corresponding cows are shown in (d).

each bounding box in t_1 to every bounding box in t_2 to compute the IoU. This comparison of overlaps between bounding boxes across frames is visualized in Fig. 3(c). Finally, the IoU values for each overlapping pair are calculated, as shown in Fig. 3(d). For clarity, only the overlaps corresponding to the correct track (i.e., the same cow across frames) are displayed. In practice, however, IoU is computed for all possible bounding box pairs between the two frames.

2.1.3. Implementation pipeline

FlowReID is designed as a targeted post-processing module rather than being executed after every detection. Instead, it is a post-processing amending step that can be executed either continuously after a certain number of frames or via some trigger or anomaly detection, for example, when the confidence of the detections drops below a certain threshold. In our case, FlowReID was triggered in post-processing if overlapping detections are present or the current number of tracks exceeds the initial number of tracks at the start of the video, and IoU overlap existed between concurrent tracks. FlowReID is subsequently applied to the subset of frames where the anomaly persists. The FlowReID protocol is implemented as an additional module within the current industry standard, YOLOv8 [2], which uses BoT-SORT for multi-object tracking [4]. Fig. 4 shows the BoT-SORT tracker pipeline [4] with the additional post-processing pipeline incorporating the BLP method.

For each selected sequence, the BLP problem is created using cost functions derived from the IoU data. Once the problem is solved, the tracks are updated by merging the re-identified sequence into the larger sequence of frames, thus correcting the mistakes that occurred within those frames. In other words, the IDs of the objects are corrected from past time steps up to the present time step.

Our implementation uses the model of (1)–(5), without the extension in (6)–(8). For this reason, one of the criteria for the frame se-

quence selection is that the number of detections in each frame must be greater than or equal to the number of detections in the first frame. Sequences for which the number of detections goes below the criterion are discarded. Implementing the model extension (6)–(8) is left as future work.

2.2. Quantum enhanced re-identification

In order to use a quantum annealer to solve the post-processing reID problem, one needs to formulate the BLP optimization problem as a QUBO. In short, a QUBO problem seeks to minimize (or maximize) a quadratic objective function defined over binary variables, without any explicit constraints other than the binary domain. Formally, the objective can be expressed as:

$$\min_{x \in \{0,1\}^n} x^T Q x, \quad (9)$$

where Q is an $n \times n$ real-valued matrix encoding both linear and quadratic interactions among variables. Despite its apparent simplicity, QUBO is NP-hard and serves as a unifying framework for numerous classical problems like the maximum cut, graph coloring, and the partition problem [27].

We need to convert the model in (1)–(5) into an unconstrained one. For this, constraints need to be transformed into penalty terms that are added to the objective value. For example, for an objective function $c^T \cdot x$ and a penalty function $H(x)$, the new objective function would read

$$c^T \cdot x + \lambda H(x)$$

where λ is a parameter. For appropriate values of this parameter λ , we can find feasible solutions just by minimizing this objective function. It is therefore critical to tune the parameter λ correctly. This holds in

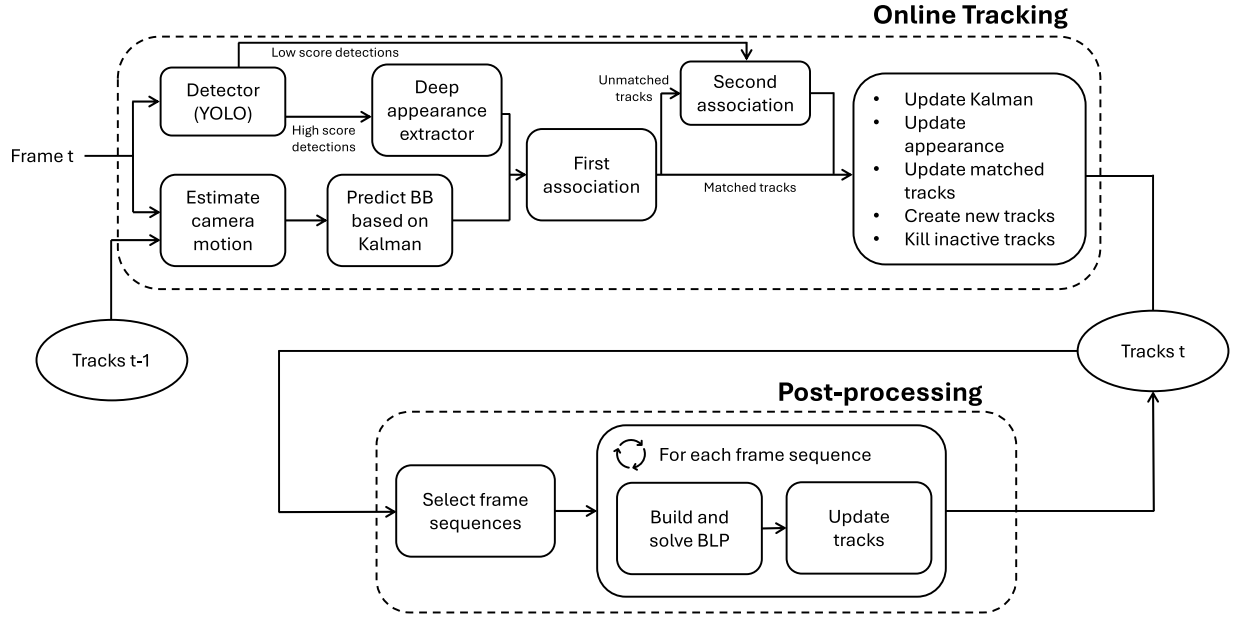


Fig. 4. Overview of the online tracking using the BoT-SORT-ReID tracker pipeline, adapted from [4], with the post-processing pipeline using our re-identification framework.

general (see e.g. the work of [27]). He determines for specific QUBO formulations the minimum value of λ such that the minimum solution is feasible. In QAOA, this is not a problem; standard IBM just takes a 'large' value. In quantum annealing, this is a problem, as tuning influences the performance: taking the value by [27] does not guarantee good results. In most cases, the penalty value must be lower.

Our model has three types of constraints, two of which are equality constraints and one of which is an inequality constraint. We treat these two types separately.

For a given vector $\mathbf{a} \in \mathbb{R}^n$ and scalar b , an equality constraint of the type $\mathbf{a}^T \cdot \mathbf{x} = b$ can be modeled via the quadratic penalty function:

$$H_1(\mathbf{a}, b, \mathbf{x}) = (\mathbf{a}^T \cdot \mathbf{x} - b)^2 = \sum_{i=1}^n \sum_{j=1}^n a_i a_j x_i x_j - 2b \sum_{i=1}^n a_i x_i + b^2. \quad (10)$$

This function takes the value zero if the constraint is satisfied and is strictly positive otherwise.

The other constraint in our model is of the type $\sum_{i=1}^n x_i \leq 1$. In other words, at most one of the elements of x_i can take the value one. This is a special type of inequality constraint that can be efficiently translated into a quadratic penalty function:

$$H_2(\mathbf{x}) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n x_i \cdot x_j. \quad (11)$$

This is, we sum over all the pairwise products of the elements of $\mathbf{x}$. The penalty function $H_2(\mathbf{x})$ is zero if the constraint is satisfied and is strictly positive if more than one element is equal to one.

Using the penalty functions defined in (10) and (11), we can reformulate each of the constraints in (1)–(5) into a penalty term, which we add to the cost function, resulting in one penalty parameter to be tuned per constraint.

2.3. Model validation

The developed reID approaches were evaluated based on different multi-object tracking metrics, comparing predictions without reID, with Ultralytics reID (used with default settings) without post-processing, and with FlowReID to ground-truth annotations for post-processing. These metrics were Higher Order Tracking Accuracy (HOTA; [28], Multi-Object Tracking Accuracy (MOTA; [29], Identification F1 (IDF1; [30], and identity switches (IDSW). Each metric has a different focus: HOTA

balances detection and association, MOTA emphasizes detection, IDF1 emphasizes association, and IDSW emphasizes the number of identity switches and losses. Metrics are presented as the average (HOTA, MOTA, IDF1) or the sum (IDSW) from all eight cameras. All algorithms were run with default parameters.

2.4. Solver comparison

In addition to overall performance, we compare different methodologies to solve the underlying minimum cost network flow problem. In particular, we use the following solvers:

- **Classical solver (CL).** We use the MILP solver of SciPy [31].
- **Simulated Annealing (SA),** with 100 reads, 500 sweeps, and the other parameters kept at their default.
- **Quantum Annealing (QA),** with 100 reads and 100 μ s annealing time on the *D-Wave Advantage2* system [32].
- **Quantum Annealing with Steepest Descent (QA + SD).** Here, we implement steepest descent as a post-processing tool. The parameters are the same as with quantum annealing.

Notice that in the latter three cases, we must perform penalty tuning. We do this for a maximum of 100 iterations, after which we stop the process and conclude that the method did not converge if no feasible solution was found. The starting value of all the penalties is $\lambda_0 = 0.1$, which is increased by a multiplicative factor $\gamma = 1.5$ at every iteration that the corresponding constraint is violated. We compare these solvers on a dataset of detection sequences extracted and identified for interest for reID through our sequence picker from two chosen videos that display ID-switching, with each sequence solved using all solvers.

The metrics we use are the following.

- **Convergence.** Whether the solver converged to a feasible solution and whether it converged to an optimal solution.
- **Total time.** The total duration of the solving process, measured end-to-end. This end-to-end time includes the full round-trip to the quantum annealer, covering both the time required to transmit the task through the cloud and the time needed to receive the returned result.
- **Execution time.** For classical and SA, this is the same as total time. For the quantum methods, we consider only QPU time and

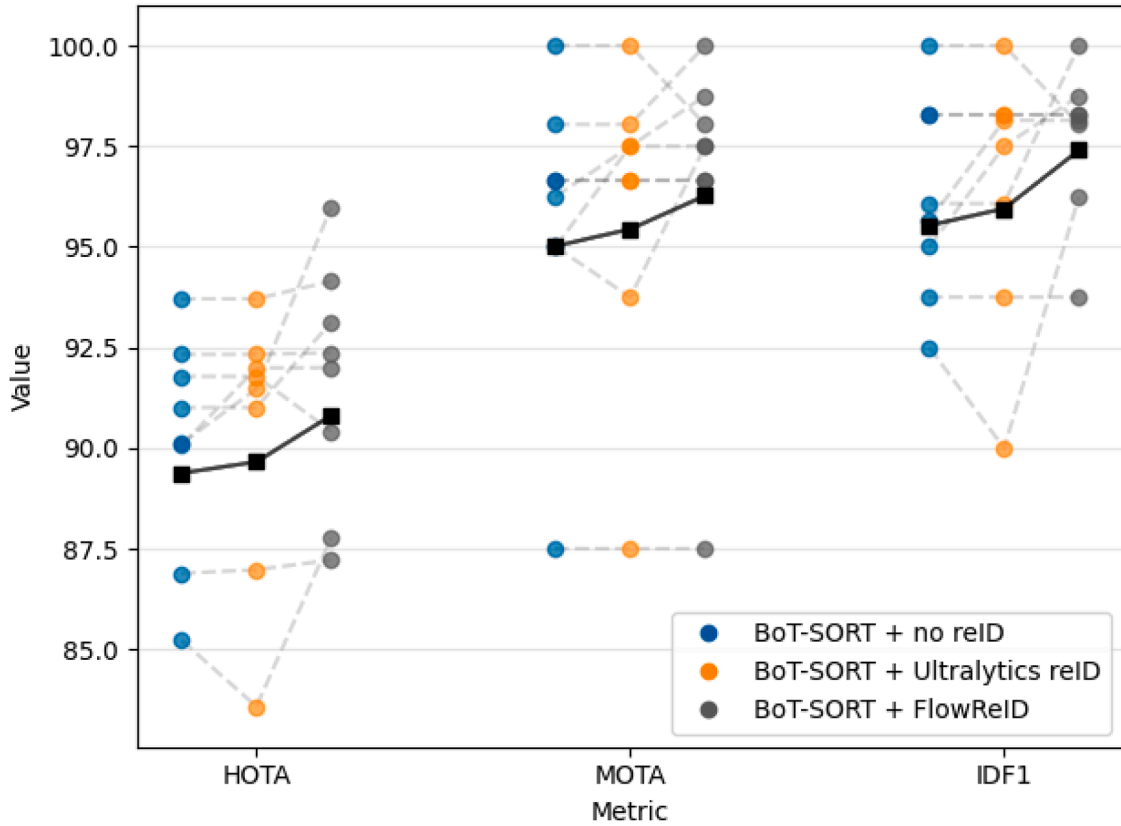


Fig. 5. Key performance metrics of multi-object tracking of the considered reID modules per sequence. Same sequences for different reID modules are connected with dashed lines. Metrics derived based on all recordings are indicated in black.

postprocessing time (if any). This excludes D-Wave connection and embedding times, which were not available separately. We note that embedding time can be a bottleneck for these methods, and therefore, in the future, quantum computing platforms should be able to provide these metrics [33].

2.5. Material

For evaluation, we use eight video recordings captured on the same day, each from a different camera, providing distinct views of a barn housing 16 Holstein Friesian cows under standard conditions. Each video recording was approximately three minutes long. Per video, five images sampled at 30-second intervals were generated and annotated with the Computer Vision Annotation Tool (CVAT; [34]), with each cow annotated with a bounding box and a consistent, unique identity across the frames and views. For model training of the initial cow detections and tracking, a panel of 390 annotated images from the same 8 views was used. Images were taken over the period of a month before the generation of the evaluation dataset.

All videos and images were recorded with Axis P1375-E cameras (Axis Communications, Lund, Sweden), using a resolution of 1920×1080 pixels at 50 frames per second. The data used in this study were collected at the Wageningen Dairy Campus as part of the Next Level Animal Sciences program.

3. Results

We first present a benchmarking experiment that aims to compare the performance of our FlowReID module to that of BoT-SORT. Second, we study the performance of the four different solvers, among which are two quantum-based ones, in solving the tracking problems that result from running the reID module.

3.1. Benchmarking of re-identification modules

On the evaluation dataset, the detection YOLO model achieved a precision of 0.98, a recall of 0.98, a mean average precision at an intersection over union of 0.5 ($mAP@0.5$) of 0.985, and a $mAP@0.5:0.95$ of 0.86. This shows that the detection model was sufficiently able to detect cows in the evaluation dataset.

Enabling reID (Ultralytics ReID and FlowReID) improved performance for all multi-object tracking evaluation metrics, showing higher HOTA, MOTA, IDF1, and fewer IDSW. The newly developed approach (FlowReID) achieved the best performance, reducing the number of identity switches from 11 or 9 with no reID or Ultralytics reID, respectively, to 5 with FlowReID, corresponding to a reduction of IDSW by 54% and 44%, respectively (Table 1). Similarly, HOTA / MOTA improved from 89.38 / 95.02 to 89.66 / 95.44 (when using Ultralytics reID) to 90.81 / 96.27 (when using FlowReID), representing relative improvements of 14% / 11% for HOTA and 25% / 16% for MOTA of FlowReID, respectively. For all but one recording, tracking metrics improved on a per-recording basis (Fig. 5). Despite this being a statistically significant improvement based on a Binomial test ($p = 0.035$), results should be interpreted cautiously because of the limited number of recordings (which also led us to not fine-tune individual components such as the used IoU based cost and remain with “standard” choices).

3.2. Benchmarking of solvers

Using our sequence picker, our FlowReID protocol was triggered 19 times across the two videos considered for the benchmarking of solvers. Regarding the performance of the different solvers, the classical solver, simulated annealing, and quantum annealing with steepest descent post-processing always reached convergence, while quantum annealing without post-processing failed to converge in 9 of the 19 samples (Table 2). The reached solution is the global optimum in all cases, except for two

Table 1

Key performance metrics of multi-object tracking of the considered reID modules.

	HOTA	MOTA	IDF1	IDSW
Bot-SORT + no reID	89.38	95.02	95.535	11
Bot-SORT + Ultralytics reID	89.66	95.44	95.95	9
Bot-SORT + FlowReID	90.81	96.27	97.40	5

Table 2

Percentage of samples that converged and percentage of samples that converged to the optimal solution, for each solver. The data shown is over the 19 samples selected for re-identification.

	CL	SA	QA	QA + SD
Converged	100%	100%	52.6%	100%
Optimal	100%	100%	52.6%	89.5%

Table 3

Average solving times for the four optimization methods. Results are shown for the 10 samples in which all methods, including QA, reached convergence and for all 19 samples. QA results are not reported for all samples because QA did not converge in 9 of the 19 samples.

		CL	SA	QA	QA + SD
Converged samples ($n = 10$)	Total time (s)	0.029	0.126	3.351	2.996
	Exec time (s)	0.029	0.126	0.059	0.056
All samples ($n = 19$)	Total time (s)	0.028	0.524	–	36.005
	Exec time (s)	0.028	0.524	–	2.624

samples when using quantum annealing with post-processing. Note that quantum annealing in these two cases did not reach convergence at all.

Regarding solving time, since quantum annealing did not converge for some samples, we can only compare solving time on the restricted set of samples where all methods converge. The classical solver required the lowest computing time (0.029 s, Table 3). Quantum annealing with and without post-processing required about twice the execution time (0.056 / 0.059 s), keeping in mind that the embedding time was not included in the total execution time, while simulated annealing took four times as long (0.126 s). We can compare the solution time of classical, simulated annealing, and quantum annealing with post-processing on all 19 samples, given that they always converge. While computing time for the classical solver is basically unchanged (0.028 s, Table 3), a sharp increase in the total solving time of simulated annealing and quantum annealing with post-processing was observed, which we attribute to a longer embedding time for the seemingly harder instances (in which QA did not reach convergence). Computing time of simulated annealing increased 4-fold to 0.524 s, while quantum annealing with post-processing took 50 times as long (2.624 s).

4. Discussion & conclusion

In this study, we investigated the application of a post-processing reID step (“FlowReID”) to extend the BoT-SORT tracking framework. The proposed approach reformulates the identity association as a binary linear program that can be solved using either classical or quantum optimization techniques. FlowReID consistently outperformed the current Ultralytics reID module across all evaluation metrics, resulting in higher overall consistency of the assigned track identity and fewer identity switches.

A key distinction between FlowReID and Ultralytics reID lies in their position within the tracking pipeline. Ultralytics ReID runs continuously during tracking, whereas FlowReID is applied as a targeted post-processing step. In addition, Ultralytics ReID uses either appearance features or geometric features (IoU) to match tracks and detections, depending on which feature could potentially produce a better match

given a minimum IoU and appearance-similarity thresholds. FlowReID only expects that spatial overlap exists and specifically targets the events where an identity switch is likely to happen.

As part of this study, we also assessed the applicability of quantum algorithms to the multi-object tracking problem. Our analysis revealed that current classical tools already operate at the desired level for this class of problems, limiting the potential impact of quantum computing in this specific scenario. This provides a meaningful insight for future research into the role of quantum computing in multi-object tracking, encouraging continued exploration with an emphasis on practical applicability.

Quantum approaches, particularly those leveraging annealing-based optimization, could offer advantages in solving problems that are NP-hard with easily verifiable solutions, referred to as NP-complete problems [27]. Therefore, while quantum computing may not outperform classical methods for the easier, small-scale instances, its potential for handling large-scale, complex configurations makes it an attractive avenue for further research and development. Research is, however, not yet conclusive on whether some of the claims of quantum advantage of lower computing time or better scaling for large-scale data are real. On this note, concerns have been raised in the literature regarding the overstatement of near-term quantum advantages and the gap between theoretical speedups and practical, hardware-constrained implementations [11,35].

As such, the results from our analysis on a relatively small dataset with a limited number of cows and duration of observation should primarily be interpreted as a proof-of-concept, in which many simplifications were adopted, such as that all cows were in the initial frame of the recording period. These assumptions limit immediate practical applicability but demonstrate the steps required to convert a classical optimization problem into a QUBO suitable for quantum annealing. The primary reason for not considering larger datasets in this study was limitations in current quantum hardware.

Note that these simplifications were primarily chosen to allow for a straightforward implementation of the quantum-based optimization pipeline, and an extension of FlowReID on traditional hardware and more robust post-processing triggers should be relatively straightforward. Consequently, the generalizability of the numerical results to larger herds or longer recording periods is not certain. As our proposed post-processing re-identification through FlowReID led to improved HOTA, MOTA, and IDF1 for all sequences except one, there should be merit in a larger-scale test of the FlowReID module.

While the use of quantum computers has previously been proposed in livestock science [12,36,37], to our knowledge, this work represents the first real-world direct application of quantum computing implemented using corresponding hardware in livestock science.

CRedit authorship contribution statement

L. Scavuzzo: Writing – review & editing, Writing – original draft, Software, Investigation, Conceptualization; **J.C. Boschero:** Writing – review & editing, Writing – original draft, Software, Investigation, Conceptualization; **J.E. Doornweerd:** Writing – review & editing, Validation, Software, Conceptualization; **J. Vandenplas:** Writing – review & editing, Conceptualization; **S.G.G. Cramer:** Writing – review & editing, Software, Investigation, Conceptualization; **E.P. Aguilera:** Writing – review & editing, Supervision, Investigation, Conceptualization; **T. Poock:** Writing – review & editing, Supervision, Project administration, Conceptualization.

Ethics declaration

Animal subject. Ethics committee approval was not required. All data used in this study were collected at the Wageningen Dairy Campus as part of the Next Level Animal Sciences program. The study was purely

observational and did not involve any experimental manipulation, intervention, or alteration of the animals' environment, management, or routine husbandry practices. The dataset consists of video recordings of 16 Holstein Friesian cows housed under standard commercial barn conditions. Recordings were obtained using fixed-position cameras and were limited to passive monitoring of normal animal behavior. No procedures were performed that could cause pain, distress, or discomfort to the animals. As the study relied exclusively on non-invasive video recordings collected under routine farm management conditions, it did not require specific ethical approval under applicable institutional or national guidelines. All data collection complied with relevant animal welfare regulations and institutional standards for the ethical use of animals in research.

Declaration of generative AI and AI-assisted technologies in the manuscript preparation process

During the preparation of this work, the authors used ChatGPT (GPT-5.2) to assist with language refinement and stylistic improvements. All content was reviewed and edited by the authors, who take full responsibility for the final version of the manuscript.

Funding statement

This work is supported by the Dutch National Growth Fund (NGF), as part of the Quantum Delta NL programme, and by the Wageningen University & Research Small Innovative Projects (WR-SIP; Project code: KB-38-001-007) that is financed by the Dutch Ministry of Agriculture, Nature and Food Quality (LNV).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

All code underlying the implementation of the QUBO optimization problem is available at <https://github.com/TNO-Quantum/problems.mot>. All video data used in this study is available upon reasonable request from Wageningen Livestock Research. Requests to access the dataset should be directed to Torsten Pook, torsten.pook@wur.nl.

References

- [1] M.H. Nidhi, K. Liu, K.J. Flay, Multi object tracking in livestock-from farm animal management to state-of-the-art methods, *Animal* 19 (5) (2025) 101503. <https://doi.org/10.1016/j.animal.2025.101503>
- [2] G. Jocher, A. Chaurasia, J. Qiu, Ultralytics YOLO (version 8.0.0): computer software, 2023, <https://github.com/ultralytics/ultralytics>.
- [3] A. Bewley, Z. Ge, L. Ott, F. Ramos, B. Upcroft, Simple online and realtime tracking, in: 2016 IEEE International Conference on Image Processing (ICIP), 2016, pp. 3464–3468. <https://doi.org/10.1109/ICIP.2016.7533003>
- [4] N. Aharon, R. Orfaig, B.-Z. Bobrovsky, BoT-SORT: Robust associations multi-pedestrian tracking, (2022). [arXiv preprint arXiv:2206.14651](https://arxiv.org/abs/2206.14651)
- [5] C. Lamping, G. Kootstra, M. Derks, Transformer-based similarity learning for re-identification of chickens, *Smart Agric. Technol.* 11 (2025) 100945. <https://doi.org/10.1016/j.atech.2025.100945>
- [6] P. Yu, T. Burghardt, A.W. Dowsey, N.W. Campbell, Holstein-Friesian re-identification using multiple cameras and self-supervision on a working farm, *Comput. Electron. Agric.* 237 (2025) 110568. <https://doi.org/10.1016/j.compag.2025.110568>
- [7] T.T. Zin, C.N. Phyto, P. Tin, H. Hama, I. Kobayashi, Image technology based cow identification system using deep learning, *Proceedings of the International Multi-Conference of Engineers and Computer Scientists* (1) (2018) 236–247.
- [8] Y. Kawagoe, I. Kobayashi, T.T. Zin, Facial region analysis for individual identification of cows and feeding time estimation, *Agriculture* 13 (5) (2023) 1016. <https://doi.org/10.3390/agriculture13051016>
- [9] N. Wojke, A. Bewley, D. Paulus, Simple online and realtime tracking with a deep association metric, (2017). <https://doi.org/10.48550/arXiv.1703.07402>
- [10] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, *ImageNet Classif. Deep Convolutional Neural Netw.* (2012) 1097–1105. <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>.
- [11] J. Preskill, Quantum computing in the NISQ era and beyond, *Quantum* 2 (2018) 79. <https://doi.org/10.22331/q-2018-08-06-79>
- [12] T. Pook, J. Vandenplas, J.C. Boschero, E. Aguilera, K. Leijnse, A. Chauhan, Y. Bouzembrak, R. Knapen, M. Aldridge, Assessing the potential of quantum computing in agriculture, *Comput. Electron. Agric.* 235 (2025) 110332. <https://doi.org/10.1016/j.compag.2025.110332>
- [13] V. Vedral, M.B. Plenio, Basics of quantum computation, *Prog. Quantum Electron.* 22 (1) (1998) 1–39. [https://doi.org/10.1016/S0079-6727\(98\)00004-4](https://doi.org/10.1016/S0079-6727(98)00004-4)
- [14] D. Risè, M.P. Da Silva, C.A. Ryan, A.W. Cross, A.D. Córcoles, J.A. Smolin, J.M. Gambetta, J.M. Chow, B.R. Johnson, Demonstration of quantum advantage in machine learning, *Npj Quantum Inf.* 3 (1) (2017) 1–5. <https://doi.org/10.1038/s41534-017-0017-3>
- [15] P.W. Shor, Algorithms for quantum computation: discrete logarithms and factoring, *Proceedings of the 35th Annual Symposium on Foundations of Computer Science* (1994) 124–134. <https://doi.org/10.1109/SFCS.1994.365700>
- [16] IBM, IBM delivers new quantum processors, software, and algorithm breakthroughs on path to advantage and fault tolerance, 2025, <https://newsroom.ibm.com/2025-11-12-ibm-delivers-new-quantum-processors,-software,-and-algorithm-breakthroughs-on-path-to-advantage-and-fault-tolerance>.
- [17] H. Neven, V. Smelyanskiy, Our quantum echoes algorithm is a big step toward real-world applications for quantum computing, 2025, <https://blog.google/innovation-and-ai/technology/research/quantum-echoes-willow-verifiable-quantum-advantage/>.
- [18] V. Hassija, V. Chamola, V. Saxena, V. Chanana, P. Parashari, S. Mumtaz, M. Guizani, Present landscape of quantum computing, *IET Quantum Commun.* 1 (2) (2020) 42–48. <https://doi.org/10.1049/iet-qt.2020.0027>
- [19] QuTech, Quantum delta NL awarded 615 million euro from Netherlands' national growth fund to accelerate quantum technology, 2021, <https://qutech.nl/2021/04/09/quantum-delta-nl-awarded-615-million-euro-from-netherlands-national-growth-fund-to-accelerate-quantum-technology/>.
- [20] E. Farhi, J. Goldstone, S. Gutmann, A quantum approximate optimization algorithm, (2014). [arXiv preprint arXiv:1411.4028](https://arxiv.org/abs/1411.4028)
- [21] P. Skrzypczyk, D. Cavalcanti, Semidefinite Programming in Quantum Information Science, IOP Publishing, 2023. <https://doi.org/10.1088/978-0-7503-3343-6>
- [22] N. Wiebe, D. Braun, S. Lloyd, Quantum algorithm for data fitting, *Phys. Rev. Lett.* 109 (5) (2012) 050505. <https://doi.org/10.1103/PhysRevLett.109.050505>
- [23] J. Zaech, A. Liniger, M. Danelljan, D. Dai, L. van Gool, Adiabatic quantum computing for multi object tracking, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022) 8801–8812. <https://doi.org/10.1109/CVPR52688.2022.00861>
- [24] Y. Ihara, Enhancing multiple object tracking accuracy via quantum annealing, *Sci. Rep.* 15 (1) (2025) 24294. <https://doi.org/10.1038/s41598-025-07492-7>
- [25] M. Hewitt, G.L. Nemhauser, M.W.P. Savelsbergh, Combining exact and heuristic approaches for the capacitated fixed-charge network flow problem, *INFORMS J. Comput.* 22 (2) (2010) 314–325. <https://doi.org/10.1287/ijoc.1090.0348>
- [26] J. Berclaz, F. Fleuret, E. Turetken, P. Fua, Multiple object tracking using k-shortest paths optimization, *IEEE Trans. Pattern Anal. Mach. Intell.* 33 (9) (2011) 1806–1819. <https://doi.org/10.1109/TPAMI.2011.21>
- [27] A. Lucas, Ising formulations of many NP problems, *Front. Phys.* 2 (2014) 5. <https://doi.org/10.3389/fphy.2014.00005>
- [28] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, B. Leibe, HOTA: a higher order metric for evaluating multi-object tracking, *Int. J. Comput. Vis.* 129 (2) (2021) 548–578. <https://doi.org/10.1007/s11263-020-01375-2>
- [29] K. Bernardin, R. Stiefelhofen, Evaluating multiple object tracking performance: the clear mot metrics, *EURASIP J. Image Video Process.* 2008 (1) (2008) 246309. <https://doi.org/10.1155/2008/246309>
- [30] E. Ristani, F. Solera, R. Zou, R. Cucchiara, C. Tomasi, Performance measures and a data set for multi-target, multi-camera tracking, *Lect. Notes Comput. Sci. Incl. Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinform.* (2016). <https://doi.org/10.1007/978-3-319-48881-3-2>
- [31] S. Developers, *scipy.optimize.milp* — SciPy v1.17.0 Manual, 2026, <https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.milp.html>.
- [32] I. D-Wave Quantum, Systems — d-wave quantum, 2026, <https://www.dwavequantum.com/solutions-and-products/systems/>.
- [33] F. Phillipson, Fair benchmarking combinatorial optimization solvers in the era of emerging computing paradigms, *Commun. Comput. Inf. Sci.* (2513) (2025) 79–93. <https://doi.org/10.1007/978-3-031-94263-1-6>
- [34] B. Sekachev, N. Manovich, A. Zhavoronkov, Computer vision annotation tool, *Zenodo* (2019). <https://doi.org/10.5281/zenodo.3497105>
- [35] T. Albash, D.A. Lidar, Adiabatic quantum computation, *Rev. Mod. Phys.* 90 (1) (2018) 015002. <https://doi.org/10.1103/RevModPhys.90.015002>
- [36] C. Maraveas, D. Konar, D.K. Michopoulos, K.G. Arvanitis, K.P. Peppas, Harnessing quantum computing for smart agriculture: empowering sustainable crop management and yield optimization, *Comput. Electron. Agric.* 218 (2024) 108680. <https://doi.org/10.1016/j.compag.2024.108680>
- [37] C. Maraveas, K.G. Arvanitis, T. Bartzanas, D. Loukatos, Potential applications of quantum sensors in agriculture: a review, *Comput. Electron. Agric.* 235 (2025) 110420. <https://doi.org/10.1016/j.compag.2025.110420>