2026 IEEE/SICE International Symposium on System Integration (SII)
January 11-14, 2026. Cancún, México

Sequence Optimization in Multi-Camera Robotic Visual Inspection

Miha Deniša, Tadej Petrič, and Aleš Ude

Abstract— **Low-volume, high-mix manufacturing presents unique challenges for visual inspection, where frequent product changes make automation difficult. Manual inspection remains common but is slow and error-prone, while fully automated systems are often too costly for small and medium-sized enterprises (SMEs). We present a method to optimize the sequence of image acquisitions in robotic visual inspection. The problem is formulated as a unidirectional weighted graph and solved using Travelling Salesman Problem (TSP) techniques. Unlike prior work focused on single-camera setups, we address the more complex case of two-camera inspection with larger numbers of inspection points, introducing a geometric grouping strategy that clusters inspection points by planar regions derived from object geometry. This enables efficient parallel use of cameras while maintaining low planning complexity. The proposed framework supports agile reconfiguration of inspection tasks, making it suitable for high-mix industrial environments. In simulation of a real-world scenario, our method reduces inspection cycle times by up to 2.35× while maintaining near-optimal sequencing, demonstrating its potential to make multi-camera robotic inspection more practical for agile manufacturing.**

I. Introduction

Automated visual inspection offers higher precision, repeatability, and speed than manual inspection, yet adoption in low-volume, high-mix (LVHM) manufacturing [1] remains limited due to cost, rigidity, and the complexity of sequence planning. Conventional systems are designed for fixed trajectories and single-camera setups, which do not suit the frequent reconfiguration and varied part geometries typical of agile manufacturing. One of the key bottlenecks is optimizing the order of image acquisitions across many inspection points (IPs)—particularly in two-camera cells—so as to minimize robot motion and cycle time without compromising coverage or image quality.

While prior work has addressed TSP-based sequence optimization or multi-camera inspection separately, few methods integrate these approaches with strategies that exploit camera placement and object geometry to reduce planning complexity. In this work, we model sequence planning for robotic visual inspection as a unidirectional weighted graph and solve it using Travelling Salesman Problem (TSP) techniques. We extend the problem from the simple single-camera case to two-camera inspection with larger sets of IPs

*This research has received funding from the program group P2-0076 Automation, robotics, and biocybernetics supported by the Slovenian Research Agency, and from DIGITOP, GA no. TN-06-0106, funded by Ministry of Higher Education, Science and Innovation of Slovenia, Slovenian Research and Innovation Agency, and European Union – NextGenerationEU.

The authours are with Dept. of Automatics, Biocybernetics and Robotics, Jožef Stefan Institute, Ljubljana, Slovenia {miha.denisa,tadej.petric,ales.ude}@ijs.si

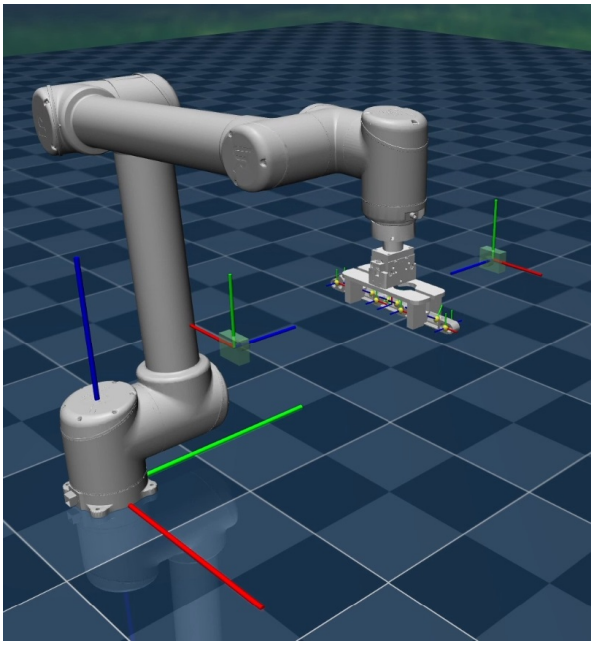

Fig. 1: The visual inspection cell in simulation. The robot holds the industrial object to be inspected by two opposing cameras.

(see Fig. 1), introducing a *plane-based grouping strategy* that clusters inspection points according to local surface geometry. This geometric grouping enables efficient parallel use of cameras while limiting the combinatorial growth of the search space. Recent advances in viewpoint planning and joint station–viewpoint optimization further motivate such geometric grouping approaches, demonstrating growing interest in efficient multi-camera sequencing [2]–[4].

Related research spans CAD-based viewpoint definition [5], [6], coverage-planning methods such as iterative viewpoint resampling [7] and adaptive viewpoint sampling [8], and TSP-formulated inspection sequencing under motion and accessibility constraints [3], [4], [9], [10]. Approaches combining discrete sequencing with continuous motion planning have shown notable cycle-time reductions, especially when adapted to industrial robot kinematics and workspace limits. Multi-camera and multi-view systems have also improved inspection robustness and coverage [11], but rarely optimize the capture order across all viewpoints. Recent studies have further advanced inspection sequencing and multi-camera coordination. Magaña *et al.* [2] proposed a geometry-driven viewpoint planning framework for range sensors using feature-cluster constrained spaces, while Kong

et al. [3] introduced a joint station–viewpoint coverage-path formulation solved via tailored genetic algorithms. Zhao *et al.* [4] presented a two-stage coverage-constrained inspection planner for structural applications. These works reinforce the relevance of geometry-aware grouping and sequencing strategies and motivate the plane-based grouping adopted in this study.

Compared with previous studies, the proposed framework explicitly considers scalability and applicability for small and medium-sized enterprises (SMEs). The grouping approach allows inspection plans to be rapidly regenerated when part geometry or camera configuration changes, improving adaptability in LVHM environments. This paper closes that gap by presenting a two-camera sequence optimization framework for LVHM inspection cells and providing a comparative evaluation of three Travelling Salesman Problem (TSP) solving strategies: a 2-opt heuristic [12], Binary Integer Programming (BIP) [12], [13], and the Held–Karp dynamic programming algorithm [14]. Each method is tested in both single- and two-camera configurations on an industrial inspection scenario, enabling a quantitative analysis of trade-offs between runtime, optimality, and achievable cycle-time savings.

II. Methodology

We denote inspection points (IPs) as predefined points on the object to be inspected. These points are defined by the user, typically on a 3D model of the object, specifying the exact locations that need to be visually inspected. Each IP is assumed to be fully captured by a single camera image, unless the area is large, in which case multiple IPs are defined to cover it. The required camera angle for each inspection can either be predefined or determined based on the surface normal vector at each IP.

We assume that all cameras in the system have the same intrinsic parameters. As a result, the viewpoint pose, which is the relative pose of the camera to the IP from which an appropriate image of the IP can be captured, remains the same across all cameras. This simplifies the process of determining the camera configuration for inspection.

The spatial relationships among the robot, its end-effector, and the camera are defined using transformation matrices. The pose of the robot's end-effector $\mathbf{T}_r$ is computed such that the camera can capture the inspection point from the correct viewpoint. This pose is obtained from the kinematic chain

$$\mathbf{T}_r = \mathbf{T}_b^{-1}\mathbf{T}_c\mathbf{T}_v\mathbf{T}_{\text{ins}}^{-1}, \tag{1}$$

where $\mathbf{T}_b$ is the robot base pose, $\mathbf{T}_c$ is the camera pose, $\mathbf{T}_v$ is the viewpoint pose, and $\mathbf{T}_{\text{ins}}$ is the inspection-point pose.

To determine the robot's joint configuration needed to achieve this end-effector pose, we apply inverse kinematics. The inverse-kinematics function f^{IK} computes the joint angles $\mathbf{q}_r$ required to reach the pose $\mathbf{T}_r$:

$$\mathbf{q}_r = f^{\text{IK}}(\mathbf{T}_r). \tag{2}$$

For scenarios with multiple inspection points, indexed by $i = 1, \ldots, m$, and multiple cameras, indexed by $j = 1, \ldots, n$, the robot configuration for each camera–IP pair is denoted as

$$\mathbf{q}_r^{(i,j)}. \tag{3}$$

Each configuration $\mathbf{q}_r^{(i,j)}$ positions the object such that the i-th IP is in view of the j-th camera.

The goal is to determine the sequence of inspection operations that minimizes the total execution time. We model this as a graph-search problem, where each possible robot configuration for capturing an inspection point is represented as a node. The corresponding robot configurations $\mathbf{q}_r^{(i,j)}$, for each inspection point $i \in \{1, \ldots, m\}$ and camera $j \in \{1, \ldots, n\}$, are represented as states $s(i,j)$ in the graph. To account for the initial pick-up of the object and its placement after inspection, an additional state $s(pp)$ is included.

We construct a fully connected, weighted, undirected graph

$$\mathcal{G} = (\mathcal{S}, \mathcal{E}, w), \tag{4}$$

where $\mathcal{S}$ is the set of all states $s(i,j)$ and $s(pp)$, $\mathcal{E}$ is the set of edges connecting every pair of states, and $w(s_1, s_2)$ is the transition time between states s_1 and s_2, i.e., the time required for the robot to move from s_1 to s_2.

Let $\mathbf{q}^{(s)} = (q_1^{(s)}, \ldots, q_{DOF}^{(s)})^\top$ denote the joint configuration at state s, where DOF denotes the number of robot joints. Assuming all robot joints share the same maximum velocity $\dot{q}_{\max}$ and acceleration $\ddot{q}_{\max}$, the motion duration is governed by the largest joint displacement between the two configurations,

$$\hat{q}_{\max}^{(s_1,s_2)} = \max_{k \in \{1,\ldots,DOF\}} \left| q_k^{(s_2)} - q_k^{(s_1)} \right|. \tag{5}$$

We adopt a trapezoidal velocity profile, under the assumption that $\ddot{q}_{\max}$ is sufficiently large and $\hat{q}_{\max}^{(s_1,s_2)}$ is not negligibly small. In this case, the transition duration is

$$d(s_1, s_2) = \frac{\dot{q}_{\max}}{\ddot{q}_{\max}} + \frac{\hat{q}_{\max}^{(s_1,s_2)}}{\dot{q}_{\max}} + \hat{q}_{\text{add}}, \tag{6}$$

where $\hat{q}_{\text{add}}$ accounts for additional factors such as settling or vision-system trigger delays.

Having defined the weighted graph $\mathcal{G}$, the problem of determining the optimal inspection sequence can be formulated as a Travelling Salesman Problem (TSP) over the set of states. In the single-camera case, the TSP is solved directly on $\mathcal{G}$ to obtain the sequence of IPs that minimizes total transition time.

Plane Grouping and Assignment: In this work, inspection points were manually assigned to planar groups according to the known geometry of the object, which consists of several flat surfaces. Each group corresponds to a distinct plane of the object that can be observed from a consistent viewing direction. For more complex geometries or larger numbers of inspection points, this assignment could be automated using geometric criteria—for example, by clustering points with similar surface normals (within about 5°) and nearby centroids (within 2–3 mm). Such an approach would

identify coplanar regions comparable to those obtained from point-cloud plane segmentation while avoiding the need for additional preprocessing.

In the two-camera configuration, the object is rigidly held by the robot, and the cameras are mounted on opposite sides of the inspection area (Fig. 1). Each inspection point must therefore be observed by exactly one camera. Directly solving a full TSP that considers every possible camera–IP combination quickly becomes intractable; instead, the planar grouping serves to constrain the search space. All IPs lying on the same plane are assigned to a single camera, ensuring that viewpoints within a plane are captured from consistent orientations and avoiding large end-effector reorientations. In the experimental object (Fig. 2), this procedure yields three planar groups, each of which can be assigned to either Camera 1 or Camera 2, giving $2^3 = 8$ possible group-to-camera assignments. For each assignment, a single TSP instance is solved, and the plan with the lowest total inspection time is selected as optimal.

In this framework, we compare three different TSP-solving strategies. The first is the *2-opt* heuristic [12], a fast local-search method that often yields near-optimal solutions. The second is a *Binary Integer Programming (BIP)* formulation [12], [13], which solves the TSP exactly using a branch-and-cut method, guaranteeing global optimality. The third is the *Held–Karp dynamic-programming algorithm* [14], a classical exact method with exponential complexity, used here as a gold-standard optimality baseline. Each method is evaluated for both the single- and two-camera configurations, enabling a quantitative comparison of runtime, optimality, and scalability.

III. Evaluation

The performance of the three TSP solution strategies—2-opt [12], Binary Integer Programming (BIP) [12], [13], and Held–Karp [14]—is assessed in the context of inspection sequencing with more than ten IPs and up to two cameras. The evaluation focuses on two aspects: (1) the trade-off between runtime and solution optimality for different TSP solvers, and (2) the effect of using a two-camera configuration with planar grouping on overall inspection time. In all two-camera tests, each IP is inspected by exactly one camera, and all $2^3 = 8$ group-to-camera assignments are evaluated by solving one TSP per assignment and selecting the best result.

Fig. 2 shows the simulated inspection object used in our experiments. The object has 14 IPs distributed across three planes. This layout allows us to test the grouping strategy described in Section II, where planar groups are assigned to one of the two cameras to minimize motion and cycle time. Different sized groups of IPs were used, $n \in \{6, 9, 12, 15\}$, to evaluate the impact of the number of IPs. Each group included IPs from all three planes of the object and the same groups were used throughout the evaluation.

For the single-camera case, the problem reduces to solving a single TSP instance over all IPs. For the two-camera case, IPs are first grouped by inspection plane, each group assigned to exactly one camera in a given plan, and all 2^3 possible assignments are evaluated. This procedure ensures that parallel execution benefits are captured while respecting the grouping constraints.

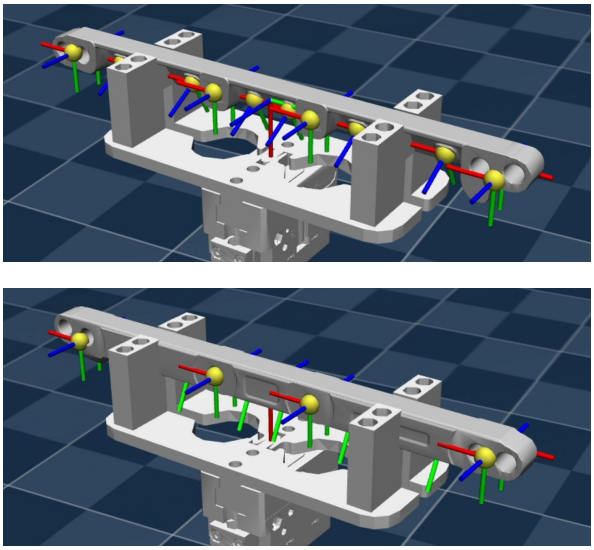

Fig. 2: A rendering of the robot gripper holding the industrial object to be inspected, shown from two viewpoints. Inspection points (IPs) are marked with yellow spheres and coordinate frames. The 14 IPs lie on three distinct planes, illustrating the planar grouping used in the two-camera experiments.

Fig. 3 shows the runtime–optimality trade-off for the single-camera case. The 2-opt heuristic achieves optimal or near-optimal tours with sub-millisecond runtimes, while both BIP and Held–Karp return provably optimal solutions. As expected, Held–Karp's exponential scaling limits its practical use to small problem sizes, whereas BIP's performance depends heavily on the number and structure of subtours encountered during optimization. Interestingly, BIP runtimes can occasionally drop as the number of IPs increases because branch-and-cut performance depends not only on problem size but also on the geometry of the distance matrix: certain configurations produce stronger initial linear relaxations and require fewer subtour-elimination constraints, reducing search time despite a larger n.

In the two-camera grouped setting (Fig. 4), 2-opt again produces near-optimal tours, with a maximum deviation of 1.3% at $n = 12$. Held–Karp remains the most expensive computationally, while BIP maintains optimality with moderate but variable runtimes. Enumerating and solving all eight group-to-camera assignments adds only minor runtime overhead for 2-opt but can impact optimal solvers more strongly depending on instance structure.

Fig. 5 compares runtimes directly between the single- and two-camera cases. The overhead from enumerating and solving all eight assignments is negligible for 2-opt but can either increase or leave runtimes unchanged for BIP and Held–Karp, depending on the geometry of the grouped instances.

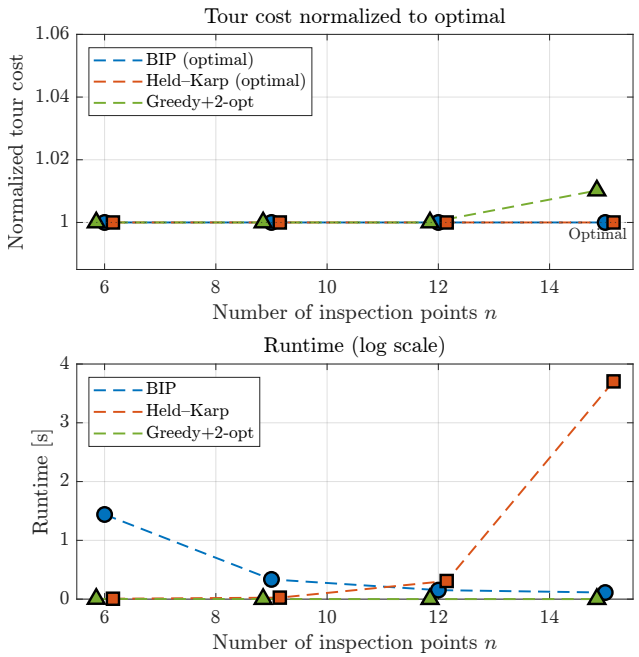

Fig. 3: Single-camera sequencing results for $n \in \{6, 9, 12, 15\}$ inspection points. Top: Tour cost normalized to the optimal (BIP/Held–Karp). 2-opt matches the optimal for $n \leq 12$ and deviates by only 1.02% at $n = 15$. Bottom: Runtime on a logarithmic scale. Held–Karp exhibits exponential scaling, 2-opt remains near-constant, and BIP runtimes vary non-monotonically due to branch-and-cut subtour-elimination dynamics.

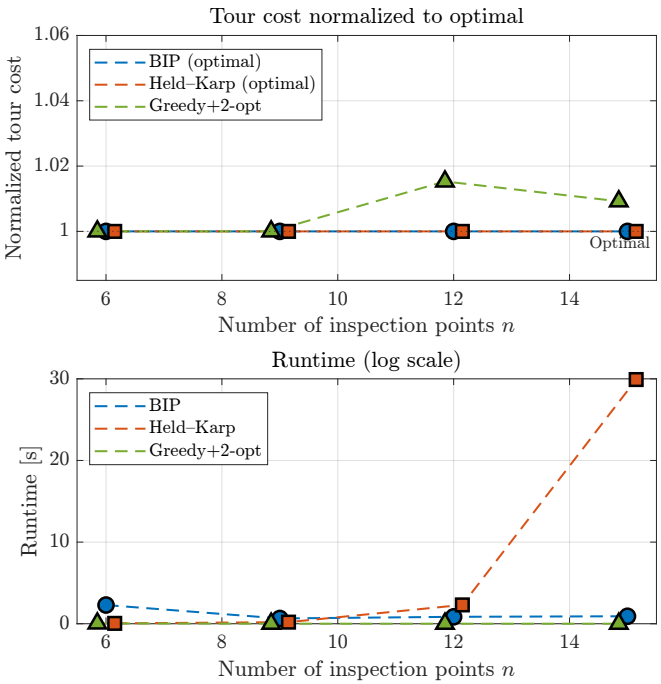

Fig. 4: Two-camera sequencing with grouped inspection planes. For each $n \in \{6, 9, 12, 15\}$, all 2^3 camera-to-group assignments are evaluated, each with exactly one camera per IP, and the best sequence is reported. Top: Normalized tour cost. 2-opt remains within 1.3% of optimal for all cases. Bottom: Runtime on a logarithmic scale, showing the same qualitative ranking as in the single-camera case.

Finally, Fig. 6 and Table I quantify the cycle-time savings achievable with a second camera. Across all tested sizes, inspection time is reduced by a factor of 1.75–2.35, with the largest gains in more compact layouts ($n = 6$). As n increases, the relative benefit declines slightly, reflecting that when points are more dispersed, parallelization is less able to offset travel time between distant groups. These results support the case for two-camera setups in low-volume, high-mix inspection cells, where substantial cycle-time reductions can be achieved with modest additional hardware complexity.

TABLE I: Optimal inspection times for single- and two-camera setups, from BIP solutions.

n	$T_{1\,\mathrm{cam}}$ [s]	$T_{2\,\mathrm{cam}}$ [s]	Improvement factor
6	6.2636	2.6643	2.351
9	7.0060	3.3878	2.068
12	7.6583	4.0326	1.899
15	8.4517	4.8193	1.754

Across all experiments, the relative ranking of solvers remains consistent: 2-opt delivers near-optimal tours at negligible computational cost, BIP achieves exact solutions with moderate runtimes and is a suitable alternative when optimality must be guaranteed, and Held–Karp serves as a gold-standard baseline limited by its exponential scaling. Introducing a two-camera configuration with grouped inspection planes yields substantial cycle-time reductions—up to a factor of 2.35—while preserving tour quality across all solvers.

Scalability and Practical Considerations

Although the evaluation considered up to fifteen inspection points, the observed solver trends indicate predictable scaling to larger instances. The 2-opt heuristic exhibits approximately quadratic growth in runtime, allowing practical use even for several dozen inspection points, whereas the Held–Karp algorithm grows exponentially and is therefore used only as a reference. Based on these trends, the framework is expected to remain computationally feasible for at least seventy IPs, which will be examined in forthcoming experiments.

In practice, the proposed two-camera configuration assumes partial field-of-view overlap sufficient to cover each inspection plane. Variations in lighting or exposure primarily influence image quality rather than sequencing performance, since motion duration dominates total cycle time. Alternative metaheuristics such as genetic algorithms or simulated annealing were also considered conceptually, but for the problem sizes typical of LVHM inspection cells they offer no clear advantage over the 2-opt heuristic, which achieves near-optimal results at a fraction of the runtime.

Future experimental validation on a physical inspection cell will assess robustness under realistic lighting and cal-

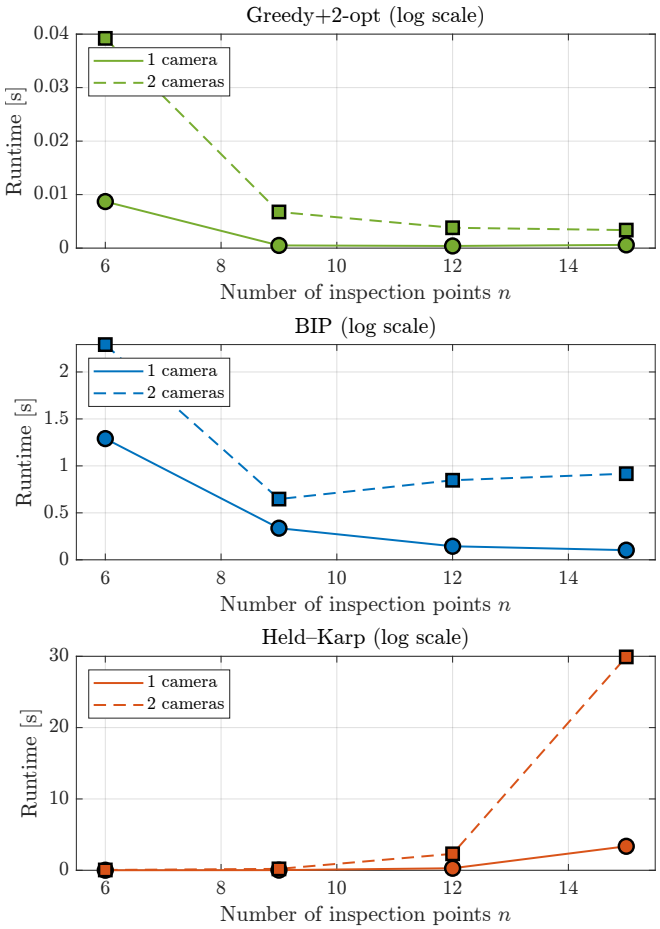

Fig. 5: Runtimes for single- (solid) vs. two-camera (dashed) sequencing across $n \in \{6, 9, 12, 15\}$. Panels show 2-opt, BIP, and Held–Karp; runtime is plotted on a logarithmic scale.

ibration conditions and verify scalability for parts with more than seventy inspection points. These extensions will further demonstrate the method's suitability for SME-scale production environments.

IV. Conclusion

This work addressed the problem of minimizing inspection cycle times in robotic visual inspection for low-volume, high-mix manufacturing, where efficiency gains directly impact production throughput. The sequencing task was modeled as a weighted graph and solved as a Travelling Salesman Problem, with three approaches compared: the 2-opt heuristic, Binary Integer Programming (BIP), and the Held–Karp dynamic programming algorithm. Experiments with both single- and two-camera configurations showed that 2-opt achieved near-optimal tour costs with negligible runtimes, while BIP and Held–Karp consistently found provably optimal solutions. Although Held–Karp is valuable as a gold-standard baseline, its exponential runtime limits practical deployment. BIP offers a viable alternative to 2-opt when an exact solution is required, with runtimes that remain manageable for the problem sizes considered here.

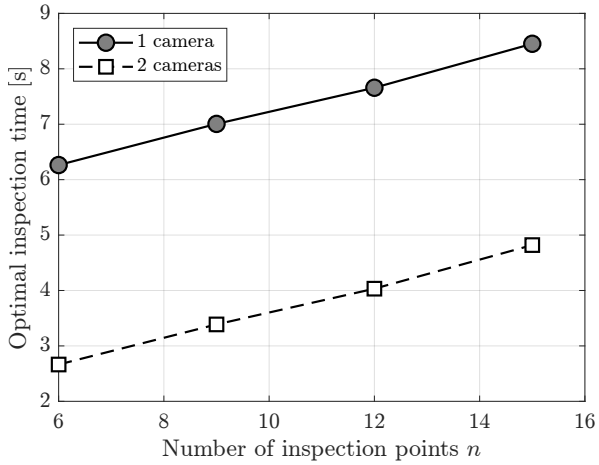

Fig. 6: Optimal inspection times from BIP solutions for single- and two-camera setups. The improvement factor is $T_{1\,\mathrm{cam}}/T_{2\,\mathrm{cam}}$, showing up to $2.35\times$ cycle-time reduction.

Introducing a two-camera setup with inspection points grouped by planar visibility reduced optimal inspection times by factors of 1.75–2.35 compared to a single-camera configuration, with the greatest benefit observed in compact inspection layouts. This demonstrates that substantial cycle-time reductions can be achieved with modest additional hardware complexity, making two-camera cells particularly attractive for agile manufacturing environments.

Future work will extend the framework to larger-scale scenarios with over seventy inspection points and more complex geometries, where automated plane detection or clustering may replace manual grouping. Additional developments will focus on adaptive object positioning—automatically adjusting pose based on image feedback to reduce glare and improve focus—and on validating the approach in a physical inspection cell under realistic lighting and calibration conditions. These efforts aim to strengthen the method's scalability and reliability, further enhancing its applicability to SME-scale robotic inspection.

References

[1] T. Gašpar, M. Deniša, P. Radanovič, B. Ridge, T. R. Savarimuthu, A. Kramberger, M. Priggemeyer, J. Roßmann, F. Wörgötter, T. Ivanovska, S. Parizi, Ž. Gosar, I. Kovač, and A. Ude, "Smart hardware integration with advanced robot programming technologies for efficient reconfiguration of robot workcells," *Robotics and Computer-Integrated Manufacturing*, vol. 66, no. 101979, 2020.

[2] A. Magaña, M. Vlaeyen, H. Haitjema, P. Bauer, B. Schmucker, and G. Reinhart, "Viewpoint planning for range sensors using feature cluster constrained spaces for robot vision systems," *Sensors*, vol. 23, no. 18, p. 7964, 2023.

[3] F. Kong, F. Du, and D. Zhao, "Station–viewpoint joint coverage path planning towards mobile visual inspection," *Robotics and Computer-Integrated Manufacturing*, vol. 91, p. 102821, 2025.

[4] Y. Zhao, B. Lu, and M. Alipour, "Optimized structural inspection path planning for automated unmanned aerial systems," *Automation in Construction*, vol. 168, p. 105764, 2024.

[5] Z. Lončarević *et al.*, "Specifying and optimizing robotic motion for visual quality inspection using cad models," *Robotics and Computer-Integrated Manufacturing*, vol. 71, no. 102136, 2021.

[6] M. FarzanehKaloorazi, I. A. Bonev, and L. Birglen, "Simultaneous task placement and sequence optimization in an inspection robotic cell," *Robotica*, vol. 39, no. 12, pp. 2110–2130, 2021.

[7] A. Bircher, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel, and R. Siegwart, "Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 6423–6430.

[8] R. Almadhoun, T. Taha, D. Gan, J. Dias, Y. Zweiri, and L. Seneviratne, "Coverage path planning with adaptive viewpoint sampling to construct 3d models of complex structures for the purpose of inspection," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 7047–7054.

[9] M. Ulrich, G. Lux, L. Jürgensen, and G. Reinhart, "Automated and cycle time optimized path planning for robot-based inspection systems," vol. 44, 2016, pp. 377–382, 6th CIRP Conference on Assembly Technologies and Systems (CATS). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212827116002389

[10] S. Song and S. Jo, "Online inspection path planning for autonomous 3d modeling using a micro-aerial vehicle," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 6217–6224.

[11] L. Niu, M. Saarinen, R. Tuokko, and J. Mattila, "Integration of multi-camera vision system for automatic robotic assembly," *Procedia Manufacturing*, vol. 37, pp. 380–384, 2019, physical and Numerical Simulation of Materials Processing IX. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2351978919312697

[12] D. L. Applegate, R. E. Bixby, V. Chvátal, and W. J. Cook, *The Traveling Salesman Problem*. Princeton: Princeton University Press, 2007. [Online]. Available: https://doi.org/10.1515/9781400841103

[13] S. D. Han and J. Yu, "Integer programming as a general solution methodology for path-based optimization in robotics: Principles, best practices, and applications," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 1890–1897.

[14] M. Held and R. M. Karp, "A dynamic programming approach to sequencing problems," *Journal of the Society for Industrial and Applied Mathematics*, vol. 10, no. 1, pp. 196–210, 1962.