Evaluation and Optimization of Dual-Arm Robot Path Planning for Human–Robot Collaborative Tasks in Smart Manufacturing Contexts

Weitian Wang
Department of Computer Science,
Montclair State University,
Montclair, NJ 07043;
Department of Automotive Engineering,
Clemson University,
Greenville, SC 29607
e-mail: wangw@montclair.edu

Yi Chen
Department of Automotive Engineering,
Clemson University,
Greenville, SC 29607
e-mail: yc4@g.clemson.edu

Yunyi Jia[1]
Department of Automotive Engineering,
Clemson University,
Greenville, SC 29607
e-mail: yunyij@clemson.edu

In human–robot collaborative tasks, the performance of robot path planning has a direct impact on the robot-to-human hand-over process, or even the collaboration quality. In this work, we propose an evaluation study on multiple robot path planners with different metrics and reveal their pros and cons in representative human–robot collaborative manufacturing contexts. Afterward, based on the proposed metrics, we define a cost function for the dual-arm robot to choose optimized path planning solutions with maximum efficiency for its human partner in human–robot collaboration. We implement the proposed evaluation and optimization approaches to multiple realistic human–robot collaborative manufacturing contexts. Experimental results and evaluations suggest that our approaches are able to provide positive solutions for the robot path planner selection and also open a window for exploring more complicated and general robot path planning applications to human–robot collaborative tasks in smart manufacturing contexts. [DOI: 10.1115/1.4046577]

Keywords: robot path planning, human–robot collaboration, smart manufacturing, cost function, human–machine interfaces, manufacturing systems, robotics

1 Introduction

Robotics technology plays a significant role in the manufacturing evolution as it can facilitate the manufacturing process to be more effective and intelligent than traditional labor-intensive manners [1–4]. Industrial robots have been extensively utilized in different kinds of manufacturing tasks such as welding, sorting, polishing, and 3D printing [5,6]. With the complexity increasing and configuration updating of products, however, the limitation of traditional robotics technology occurs gradually. For example, in the automotive final assembly phase, about 60% of tasks are manually accomplished by human workers because fenced industrial robots are inflexible and too large for automotive final assembly processes. Assembly is the most challenging subtask in vehicle production processes due to its nature of heavy manual labor, which affects the manufacturing quality and ergonomics distinctively. Such similar issues also exist in other application areas such as 3C industry, agriculture, and food industry [7,8]. In recent years, collaborative robotics technology provides effective solutions to these challenges. Collaborative robots are able to work with humans as partners to conduct shared tasks. Through human–robot collaboration, the tasks can be co-finished appropriately by taking advantages of the intelligence of humans and the accuracy of robots.

In human–robot collaborative tasks, handing over the parts or tools to each other is an indispensable and significant physical interaction [9]. Hand-over is known as a cooperative action/behavior to meet the needs of each other by delivering a desired object [10]. For example, in the automotive maintenance process, two human workers may hand over auto parts or tools to each other for the vehicle inspection or repairing. With the increasing applications in different kinds of manufacturing contexts, the collaborative robot is also expected to grasp and hand over target parts to its human partner seamlessly in human–robot collaboration just like human–human hand-over.

During the robot-to-human hand-over process, the performance of robot path planning has a direct impact on the hand-over execution, or even the collaborative task quality. Imaging that a collaborative robot grasps a target part and delivers to its human partner on a conveyor-type assembly line, if the robot plans its motion with too many waypoints or a long path length from the start-point to the destination, the human may feel uncomfortable or the human–robot team may miss the part assembly. In addition, for different sorts of tasks in diverse working settings, the robot may need different kinds of corresponding motion planning algorithms to achieve robust path planning and stable operation performance to keep the collaborative tasks well conducted. Motivated by these issues in human–robot collaboration, in this work, we develop an optimization strategy for the robot path planning by evaluating and discussing 12 kinds of path planning algorithms in multiple types of representative human–robot collaborative tasks. This study also opens a window for exploring more complicated and general robot motion planning applications to human–robot collaborative tasks in smart manufacturing contexts.

The major contributions of this work are (1) we develop an optimal path planning selection strategy for the dual-arm robot to grasp and deliver desired parts with maximum efficiency for its human partner in human–robot collaboration and (2) we conduct a comprehensive evaluation study on multiple robot path planning algorithms with different metrics and reveal their pros and cons in typical human–robot collaborative manufacturing contexts.

2 Related Work

With the increasing employment of robotics technology in a variety of fields over the past decades, robot motion planning issues have been attracting extensive attention from both academia and industries. Such issues can be generally formulized as searching

[1] Corresponding author.

Manuscript received October 24, 2019; final manuscript received February 15, 2020; published online March 4, 2020. Assoc. Editor: Kam K. Leang.

Table 1 Robot motion planners

Planner	Ref.	Directionality	Query type	Expansion manner	Lazy
RRT	[15]	Unidirectional	Single	Random	No
FMT	[16]	Unidirectional	Single	Structured	No
PRM	[17]	Unidirectional	Multiple	Random	No
RRT*	[18]	Unidirectional	Single	Random	No
RRT-connect	[11]	Bidirectional	Single	Random	No
PRM*	[18]	Unidirectional	Multiple	Random	No
EST	[19]	Unidirectional	Single	Structured	No
SBL	[23]	Bidirectional	Single	Structured	Yes
SPARS	[20]	Unidirectional	Multiple	Structured	No
LazyPRM	[24]	Unidirectional	Multiple	Random	Yes
BKPIECE	[25]	Bidirectional	Single	Structured	No
T-RRT	[21]	Unidirectional	Single	Random	No

the task workspace of one or more complex geometric bodies for a collision-free trajectory, which connects the given start-point and target-point in obstacle-constrained or obstacle-free working environments [11,12]. The research of robot motion planning emerged in the mid-1960s and stepped into a prosperity period along with the contributions of Lozano-Perez on spatial planning [13]. Currently, the family of robot motion planner has dozens of members [14]. In this work, to investigate and optimize the robot motion planning strategy in different kinds of human–robot collaborative tasks, we study and discuss 12 kinds of typical robot motion planners. As presented in Table 1, these motion planners are categorized by different categories.

When searching paths in configuration spaces, some planners generally build one exploring tree to expand and generate a valid route from the start-point to the destination. They are also known as unidirectional motion planners, such as RRT, FMT, PRM, EST, and SPARS. In order to improve searching performance, several planners, including RRT-connect, SBL, and BKPIECE, explore paths via the bidirectional search manner by constructing two trees rooted at the start and goal configurations. These trees expand paths simultaneously, and an efficient route will be generated when they are connected.

The query types of these path planners contain single query and multiple query. In single query based planners, the heuristic algorithm generally generates the valid collision-free path by adding samples to the search space incrementally [22]. For multiple query based planners (e.g., PRM and SPARS), they usually construct roadmaps for the entire workspace. The valid configuration states, which are connected to neighboring states by edges, are sampled in the configuration space during the roadmap construction process.

In path searching processes, these planners can expand sample spaces by the random manner or the structured manner. To decrease the computation time of path selection, several planners (e.g., SBL and LazyPRM) employ the lazy collision checking strategy, by which the valid collision-free path is able to be decided from all sampled points [23–25]. As listed in Table 1, each of these path planners has its own advantages and disadvantages in robot motion planning because of their different features. Therefore, for different situations in human–robot collaboration, the best robot path planner should be selected to reduce robot motion time and improve robot motion stability.

3 Mathematical Modeling

3.1 Robot–Human Hand-Over Problem Formulation. In robot–human hand-over processes, human hand-over intentions (e.g., "I need something") can be characterized by multiple methods such as vision systems [26] and natural language processing [27]. In this study, we employ a practical and easy-to-use approach by using a wearable sensory system to parameterize human gestures to have the robot understand human hand-over intentions. The sensory system we selected is Myo, which is configured by a 9-degree-of-freedom inertial measurement unit (IMU) and 8 electromyography (EMG) sensors and can be worn by the human at his/her forearm [28,29]. The human gesture information, including forearm muscle activities and rotation angles, are able to be tracked and detected by the EMG sensor and the IMU, respectively.

Based on our previous study in Ref. [28] and the collected human gesture information, we develop a human intention recognition approach via the hidden Markov model (HMM) for human–robot collaboration in the object deliver or picking processes. The HMM can be normally parameterized by five elements

$$\lambda = \{N, M, \pi, A, B\} \tag{1}$$

where N is the number hidden states, M is the number of observation symbols, π is the initial state distribution, A is a $N \times N$ matrix for the state transition probability distributions, and B is a $N \times M$ matrix for the observation symbol probability distributions [30].

When the human needs something from the robot, human intentions, which maneuver the robot-to-human hand-over procedures, can be considered as observation states of the HMM. Each intention is described with M ($M = 11$ in this study) observation symbols, which include three forearm rotation angles and eight EMG signals collected by the Myo system. In this study, N is initialized as 30 based on the validation set approach [31]. The initial state distribution π is able to start with any hand-over intentions. The probability matrices A and B are decided by the training data of different human intentions. In the HMM training process, the goal is to adjust and evaluate the parameters in Eq. (1) to obtain the maximum probability of the human intention characterized by the observation state.

By taking advantage of the trained HMMs, we input the given human gesture information to each HMM, then we can obtain the output probabilities of this intention. Therefore, the human hand-over intention is able to be classified and matched by the corresponding HMM, which has the maximum output probability. After that, using the recognized human intention, the robot can be controlled to pick up objects and deliver them to its human partner. Since it is not the main contribution of this study, more details of human hand-over intention recognition process via HMMs can be found in our previous work [28].

3.2 Evaluation Strategy of Path Planning Algorithms. In order to comprehensively evaluate robot path planners in Table 1 in different kinds of human–robot collaboration contexts, we employ three metrics, including path length, computation time, and execution time, for them when they plan the robot from the start-point to the destination.

When the planer searches a valid path for the robot, a bunch of waypoints will be planned between two configurations. Based on the generated waypoints on the valid path, we can estimate the path length by

$$L = \sum_{n=1}^{N} \sqrt{(x_n - x_{n-1})^2 + (y_n - y_{n-1})^2 + (z_n - z_{n-1})^2} \tag{2}$$

where (x, y, z) denotes the position of the waypoints in the 3D workspace, $n = 1$ represents the start waypoints, and N is the number of waypoints. Generally, longer path length means that the planner presents lower performance in the given task. Additionally, after a given path planner runs multiple times with the same task in the workspace, the standard deviation of these path lengths is able to reflect the stability of this path planner. A high standard deviation indicates that the corresponding path planner has a low stability, while a low standard deviation indicates the inverse. Therefore, we also employ the standard deviation of the path length testing results in a certain planning task to evaluate the performance of the planner. The standard deviation can be calculated by

$$\sigma_L = \sqrt{\frac{1}{M}\sum_{m=1}^{M}\left(L_m - \frac{1}{M}\sum_{m=1}^{M}L_m\right)^2} \tag{3}$$

where M is the times of running of the path planner in the workspace, m denotes each time of running, and $m \in [1, M]$. Usually, the planner is regarded with better performance if it has a lower standard deviation of the path length testing results.

The computation time refers to the total time cost of configuration space searching, path optimization, and valid path generation when the planner is called to plan a path for the robot from a given start-point to the destination. The computation time can be detected by our simulation system (see Sec. 4). In this study, we utilize T_C to denote the computation time of a certain path planner. Similarly, the standard deviation of computation time is also used to assess the planner performance

$$\sigma_{T_C} = \sqrt{\frac{1}{M}\sum_{m=1}^{M}\left(T_{Cm} - \frac{1}{M}\sum_{m=1}^{M}T_{Cm}\right)^2} \tag{4}$$

where M is the times of running of the path planner in the workspace, m denotes each time of running, and $m \in [1, M]$. It also can be considered that the shorter of the computation time and the lower of the standard deviation, the better of the planner performance.

The execution time is known as the time cost when the robot moves from a given start-point to the destination on the generated optimized path by the planner. The execution time can be described as

$$T_E = \sum_{n=1}^{N}(T_n - T_{n-1}) \tag{5}$$

where N is the number of waypoints. We also employ the standard deviation of the execution time to depict the performance of the path planner

$$\sigma_{T_E} = \sqrt{\frac{1}{M}\sum_{m=1}^{M}\left(T_{Em} - \frac{1}{M}\sum_{m=1}^{M}T_{Em}\right)^2} \tag{6}$$

where M is the times of running of the path planner in the workspace, m denotes each time of running, and $m \in [1, M]$. Suppose the robot moves with the same velocity on the valid paths generated by different kinds of planners, the planner, which corresponds to a shorter execution time and a lower standard deviation of execution time, has better performance than others.

3.3 Optimization of Dual-Arm Robot Path Planning.

In different kinds of obstacle-free or obstacle-constrained task environments, different path planners present diverse performance. In order to enable the dual-arm robot to select a correct arm with the best path planner to collaborate with its human partner in hand-over tasks, we propose a cost function considering the planners' path length, computation time, and execution time for the robot. The cost function is defined as

$$\begin{aligned} C(L', T'_C, T'_E, S, \sigma_L, \sigma_{T_C}, \sigma_{T_E}) = {} & W_A*(L' + T'_C + T'_E) \\ & + W_B*(\sigma_L + \sigma_{T_C} + \sigma_{T_E}) \end{aligned} \tag{7}$$

where W_A is the static constant, W_B is the dynamic constant, $W_A + W_B = 1$, L', T'_C, and T'_E are mean normalized path length, computation time, and execution time.

In the robot arm and path planner selection process, the static constant W_A works as the weight of the attention that humans pay to the robot's separate running result or short-term running performance, while the dynamic constant W_B acts as the weight of the attention that humans pay to the robot's long-term operation performance. The weights W_A and W_B provide a lot of flexibility for tuning the motion planning preferences in different applications. If we alter the weights W_A and W_B in human–robot collaborative tasks, some certain features will become comparatively advantageous for different performance objectives to reflect the different preferences of the applications. For example, if the human needs to work with the robot for a same all-day collaboration task, he/she may adjust W_B to be greater than W_A. On the contrary, if the human wants to work with the robot for some separate tasks in a short term, he/she can set W_A with a bigger value than W_B. The min-max normalization method [32] is employed in this study.

Note that if a given path planner cannot generate a valid path solution for the robot in the allocated task, we will regard this path searching as failure, which will affect the value of cost function. When calculating L', T'_C, T'_E, σ_L, σ_{T_C}, and σ_{T_E} in this study, we set L, T_C, and T_E as zero in case the given planner fails to search a valid path for the robot.

According to the definitions of L, T_C, and T_E above, it can be concluded that the less of the cost function value the better of the planner performance. By leveraging the cost function, the robot can obtain the optimized selection of arm and path planner for motion planning in different kinds of task settings by

$$(R_A^*, P^*) \leftarrow \underset{R_A,\, P \in \mathbb{Q}}{C_{\min}}(L', T'_C, T'_E, \sigma_L, \sigma_{T_C}, \sigma_{T_E}) \tag{8}$$

where R_A^* denotes the kind of robot arm (e.g., left arm or right arm), P denotes the kind of planner (see Table 1), and $\mathbb{Q}$ represents the configuration space in human–robot collaborative tasks.

4 Experimental Study

4.1 Experimental Platform.

In this work, we implement and verify the proposed approach on a multi-modal collaborative assembly platform, which is structured by our lab for the study of human–robot collaboration in smart manufacturing contexts. The platform consists of a dual-arm robot YuMi and a set of multi-modal based human–robot interactive interfaces, such as wearable sensory systems, 3D vision systems, and physiological sensing systems. All the equipment of the platform are controlled by the robot operating system (ROS) [33]. To get an objective evaluation for path planners, we employ the same computer as a control center to run high level robot control algorithms and motion planning algorithms in the same working condition.

4.2 Task Description.

In this experimental study, the robot works with its human partner to assemble a high-fidelity model vehicle in the manufacturing context. The human wears the Myo sensory system to trigger the robot for hand-over tasks based on our human intention recognition model (see Sec. 3.1). The robot's motion paths are generated by the planners listed in Table 1. The Open Motion Planning Library [34] is employed to operate path planners in the ROS environment.

As presented in Fig. 1, during the robot-to-human hand-over process, the robot moves its left arm or right arm from point A_1 or

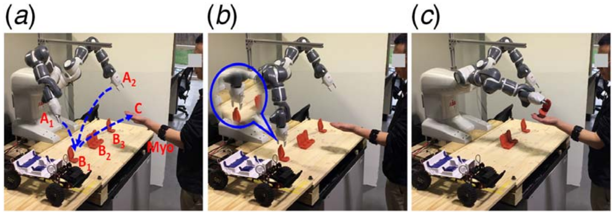

Fig. 1 The robot collaborates with its human partner for assembling a high-fidelity model vehicle: (*a*) before picking up, (*b*) picking up, and (*c*) delivering

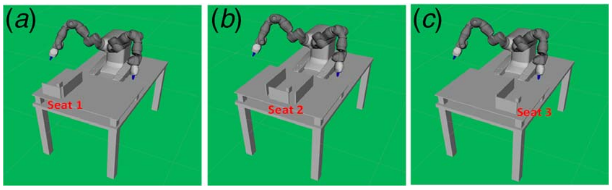

Fig. 2 Hand-over tasks in obstacle-constrained environments

A_2 to point B_1, $B_{2,}$ or B_3, then picks up the vehicle seat and delivers it to the human at point c. When the robot grasps the vehicle seat at point B_1, $B_{2,}$ or B_3, as shown in Fig. 1(*b*), its grasping orientation is assigned to be downward. The initial positions of each arm are same for each picking up-and-delivering experiment. The path length, computation time, and execution time of each experiment include the task of moving the arm from the initial position, picking up the target object, and delivering to the human. In order to obtain reliable data of the length path, computation time, and execution time, each path planner run 20 times for the allocated motion task on our experimental platform. In robot-to-human hand-over tasks, three vehicle seats (front right seat *(seat 1)*, rear seat *(seat 2)*, and front left seat *(seat 3)*) on the workbench need to be picked up and delivered by the robot. The front right seat is located near the robot right arm, the rear seat is located between the right arm and left arm, and the front left seat is located near the robot left arm. In the hand-over process, we develop a real-time robot control system using MoveIt! package [35] based on the ROS environment to acquire the number of way points, the computation time, and the execution time of each path planner. After that, the metrics we defined in Sec. 3.2 can be calculated via these collected data.

To comprehensively test and evaluate the path planners in different working environments, we employ three typical kinds of hand-over scenarios in obstacle-constrained environments. As shown in Fig. 2, different types of obstacles are arranged for each vehicle seat. In these tasks, via the generated paths from path planners, the robot utilizes its right arm or left arm to pick up the front right seat, front left seat, and rear seat, then delivers them to the human, respectively. Through the collision-aware plug-in in our simulation system, the path planner can be aware of the geometrical information of obstacles. Then, the path planner attempts to search a valid path for the robot based on the obstacle-constrained configuration space. By taking advantage of the valid paths, the robot can employ its arms to pick up vehicle seats and work with its human partner in each scenario.

5 Results and Evaluations

5.1 Path Planning Evaluation in Obstacle-Constrained Environments. The path length calculation of each path planner for the vehicle seat picking up and delivery are shown in Figs. 3 and 4, respectively. When the robot operates the seats via its right arm, as presented in Fig. 3, different path planners generate diverse lengths of valid paths. The path lengths of seat 1 operation are generally shorter than those of seat 3 operation because seat 1 is closest to the right arm. In this set of evaluation, the PRM planner shows a shortest path for seat 1 operation, while the RRT-connect planner generates a shortest path for seat 3 operation. When operating seat 2, the RRT* planner presents the best performance in path length evaluation. Similarly, in Fig. 4, when the vehicle seats are operated by the robot left arm, all the planners generate shorter valid paths for seat 3 operation than those for seat 1 operation since the seat 3 is located near the robot left arm. Among these planners, the RRT-connect generate a shortest path for the farthest seat (seat 1), while it presents a longest path for the closest seat (seat 3).

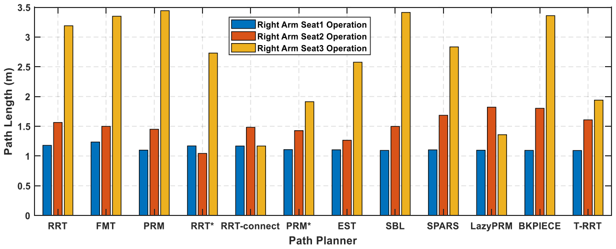

Fig. 3 The path length calculation of each path planner when the robot right arm is used

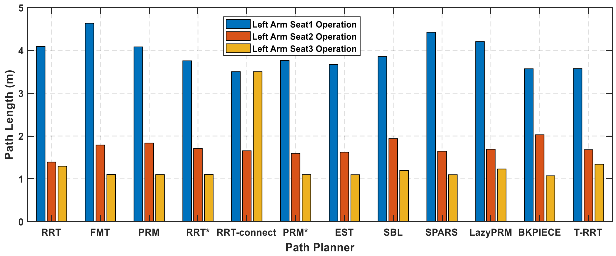

Fig. 4 The path length calculation of each path planner when the robot left arm is used

Based on the collected path lengths of each path planner, as listed in Table 2, we calculate their standard deviations. When the vehicle seats are operated using the robot right arm, it can be seen that the T-RRT planner, the FMT planner, and the LazyPRM planner present lowest standard deviations for seat 1 operation, seat 2 operation, and seat 3 operation, separately. However, the performance of these planners is different when the robot left arm is employed. In this situation, the SPARS planner shows a lowest standard deviation for seat 1 operation, while the RRT-connect planner and the FMT planner are more stable in seat 2 operation and seat 3 operation, respectively.

The computation time of each path planner in vehicle seat operation are calculated and presented in Figs. 5 and 6. When the robot right arm is employed, as shown in Fig. 5, the RRT-connect planner is able to search a valid path for seat 1 operation with the least computation time. For seat 3 operation, the EST planner shows a best performance in computation time cost. When operating seat 2, most planners' computation time is under 0.4 s. However, the BKPIECE planner, which costs about 1.1 s for searching a valid path, shows a worst performance in computation time evaluation.

The standard deviations of computation time of each path planner are shown in Table 3. When the robot right arm is used for the vehicle seat operation, the EST planner presents lowest standard deviation for seat 1 operation and seat 3 operation. For seat 2 operation, the

Table 2 The standard deviations of path length of each path planner

	Robot right arm			Robot left arm		
Planner	Seat 1	Seat 2	Seat 3	Seat 1	Seat 2	Seat 3
RRT	0.31559	0.25442	1.3649	0.94348	0.4084	0.45455
FMT	0.36609	**0.21842**	1.5142	0.57614	0.35854	**0.20558**
PRM	0.22254	0.22583	1.9171	0.85262	0.44564	0.21466
RRT*	0.31179	0.25234	1.2331	1.1206	0.48685	0.21756
RRT-connect	0.30841	0.35592	0.30841	1.0102	**0.29516**	1.0102
PRM*	0.22772	0.39028	0.54806	1.1409	0.33726	0.21466
EST	0.22062	0.30953	1.2499	0.94096	0.37805	0.21985
SBL	0.22087	0.343	1.4525	1.4369	0.67286	0.36555
SPARS	0.22099	0.473	1.3347	**0.42769**	0.65154	0.21984
LazyPRM	0.22625	0.7674	**0.089731**	0.79646	0.42261	0.45646
BKPIECE	0.22121	0.76357	1.5636	1.0474	0.85334	0.21664
T-RRT	**0.22027**	0.50056	0.51778	1.0113	0.3422	0.47436

Note: The bold values indicates the lowest standard deviations of path length of each path planner when the three seats are operated by each robot arm.

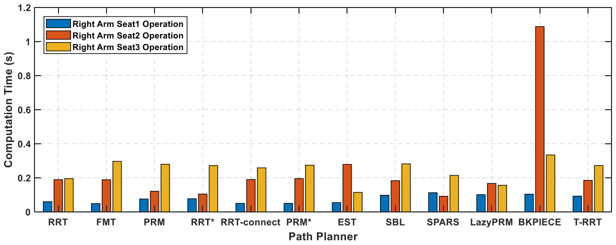

Fig. 5 The computation time calculation of each path planner when the robot right arm is used

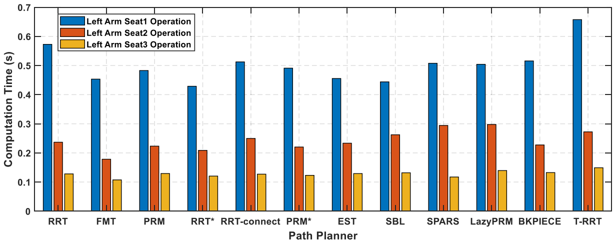

Fig. 6 The computation time calculation of each path planner when the robot left arm is used

RRT* planner shows a distinct stable when searching a valid path. When the vehicle seats are operated through the robot right arm, it can be observed that the PRM* planner, the PRM planner, and the RRT* planner have lowest standard deviations for seat 1 operation, seat 2 operation, and seat 3 operation, respectively.

Figures 7 and 8 depict the execution time evaluation of each path planner in vehicle seat operation. As presented in Fig. 7, when the robot right arm is utilized for seat 1 (the closest seat) operation, the execution time on most generated paths are similar to each other. However, for the farthest seat (seat 3), the diversity of the execution time is shown obviously. Similarly, in Fig. 8, it can be seen that the execution time of seat 3 operation are close to each other when the robot right arm is employed. When operating seat 2, the RRT planner shows a fastest path for the robot. However, the RRT-connect planner generated a most time-saved path for the robot when seat 1 need to be picked up and delivered.

The standard deviations of execution time of each path planner are illustrated in Table 4. When the robot right arm is employed to pick up and deliver vehicle seats, the FMT planner generates the most stable path than that of other planners in seat 1 operation and seat 2 operation. The LazyPRM planner presents a lowest standard deviation of execution time in seat 3 operation. When operating seat 1 and seat 2 using the robot left arm, the RRT planner shows the most robust performance. However, the PRM* is most time-saved option in seat 3 operation.

5.2 Path Planning Optimization in Different Tasks.

In this study, since we consider the proposed approach to be used in a long time human–robot collaborative work, the values of W_A and W_B in Eq. (7) are selected as $W_A = 0.4$ and $W_B = 0.6$, respectively. Based on the evaluation results in Sec. 5.1, we figure out the cost comparisons of the two robot arms in vehicle seat operation. As shown in Fig. 9, when seat 1 is operated, it can be seen that it is better to choose robot right arm than left arm. In addition, when the right arm is employed, it is the best to select the RRT planner for the robot path planning. In addition, Fig. 10 suggests that, when seat 2 is operated, the robot right arm with FMT planner is the best option. Moreover, from Fig. 11, it can be observed that the robot left arm with the T-RRT planner is the best way to operate seat 3.

The results of robot path planning optimization based on the cost function are tested and verified in realistic human–robot collaborative manufacturing contexts. Based on our developed approach in Ref. [21] and the use of standard human gesture expressions in the hand-over experiments of this study, all the human "Need" intentions could be always recognized.

Figure 12 presents one of the optimized strategies that the robot employs its right arm with the FMT planner to pick up seat 2 and deliver it to its human partner. The path length of the seat 2 operation is about 1.4 m, the computation time is about 0.19 s, the execution time is about 17 s, and the cost of is about 1.25. As shown in Fig. 12(*a*), the robot is ready for the collaborative task in the workspace. Based on the optimized strategy, the robot moves to the target seat in Figs. 12(*b*) and 12(*c*). After picking up the seat, as presented in Fig. 12(*d*), the robot hands it over to the human. Then it returns to its original position, as depicted in Figs. 12(*e*) and 12(*f*). The successful robot-to-human hand-over verification indicates that our proposed evaluation and optimization approaches are able to provide positive strategies for the robot path planner selection in human–robot collaborative tasks in smart manufacturing contexts.

6 Conclusions and Future Work

In this work, we have proposed an evaluation study on multiple robot path planners with different metrics, such as path length, computation time, and execution time, for human–robot collaboration in smart manufacturing contexts. Based on the proposed metrics, we have defined a cost function for the robot to choose optimized path planning solutions to finish corresponding human–robot collaborative tasks. Additionally, by analyzing the empirical study data, we have successfully enabled the robot to select best ways

Table 3 The standard deviations of computation time of each path planner

	Robot right arm			Robot left arm		
Planner	Seat 1	Seat 2	Seat 3	Seat 1	Seat 2	Seat 3
RRT	0.005732	0.015693	0.023198	0.25471	0.016639	0.008628
FMT	0.021055	0.012457	0.049019	0.1497	0.015096	0.008613
PRM	0.010929	0.016543	0.054537	0.14019	**0.010532**	0.010194
RRT*	0.009488	**0.006864**	0.037279	0.10403	0.057555	**0.006146**
RRT-connect	0.005826	0.013073	0.027142	0.28881	0.061862	0.016388
PRM*	0.005362	0.011472	0.022622	**0.10101**	0.084969	0.016666
EST	**0.005355**	0.23394	**0.018561**	0.15892	0.036946	0.017777
SBL	0.012124	0.007957	0.045608	0.12573	0.081813	0.017965
SPARS	0.009649	0.01134	0.021244	0.12442	0.10993	0.015761
LazyPRM	0.016859	0.026287	0.025181	0.15923	0.1004	0.010729
BKPIECE	0.008875	0.18301	0.062276	0.31677	0.074343	0.011531
T-RRT	0.006623	0.006876	0.03778	0.47941	0.10099	0.008139

Note: The bold values indicates the lowest standard deviations of path length of each path planner when the three seats are operated by each robot arm.

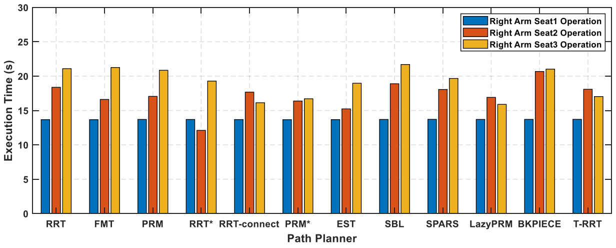

Fig. 7 The execution time calculation of each path planner when the robot right arm is used

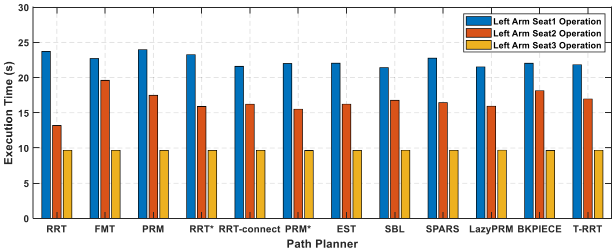

Fig. 8 The execution time calculation of each path planner when the robot left arm is used

Table 4 The standard deviations of execution time of each path planner

	Robot right arm			Robot left arm		
Planner	Seat 1	Seat 2	Seat 3	Seat 1	Seat 2	Seat 3
RRT	0.072719	3.237	2.6535	**1.8277**	**2.6631**	0.11149
FMT	**0.070287**	**0.87734**	5.0062	2.0032	3.8904	0.10948
PRM	0.10668	1.2522	3.9965	2.155	3.648	0.077358
RRT*	0.097806	4.1701	3.3931	2.5619	3.8271	0.073067
RRT-connect	0.10967	1.8574	1.8178	2.2241	3.8028	0.10638
PRM*	0.071756	2.7123	2.6167	3.4198	2.9254	**0.003147**
EST	0.10915	3.7178	2.7401	2.4048	3.2897	0.069727
SBL	0.076881	1.312	3.929	3.0558	3.2197	0.074702
SPARS	0.10657	3.4959	4.4495	2.0517	3.6225	0.07086
LazyPRM	0.077441	4.5752	**1.1507**	2.9167	4.8777	0.11208
BKPIECE	0.072314	10.599	3.9864	3.0589	3.9745	0.005566
T-RRT	0.10743	4.1671	2.7768	2.2605	4.8966	0.080472

Note: The bold values indicates the lowest standard deviations of path length of each path planner when the three seats are operated by each robot arm.

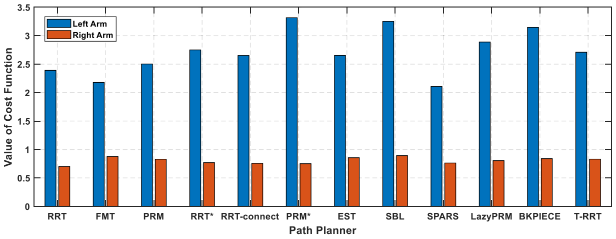

Fig. 9 Cost comparisons of two robot arms with different kinds of path planners when seat 1 is operated

to finish its tasks in different kinds of scenarios. Moreover, we have implemented the proposed evaluation and optimization approaches to multiple realistic human–robot collaborative manufacturing contexts. Experimental results and evaluations have suggested that our approaches can provide positive solutions for the robot path planner selection.

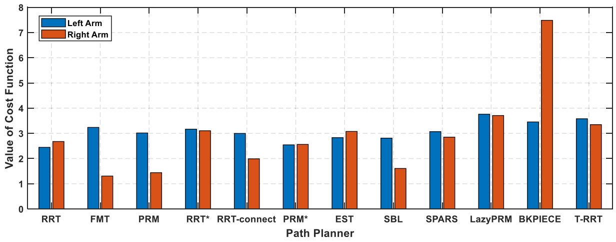

Fig. 10 Cost comparisons of two robot arms with different kinds of path planners when seat 2 is operated

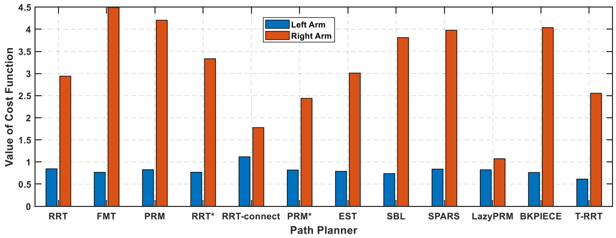

Fig. 11 Cost comparisons of two robot arms with different kinds of path planners when seat 3 is operated

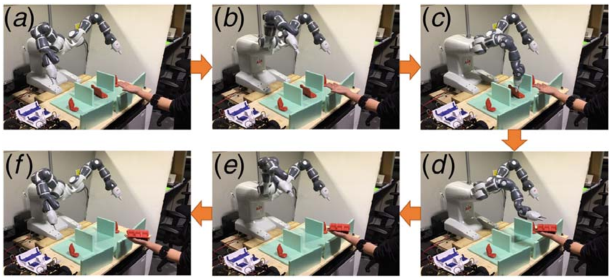

Fig. 12 Testing in realistic manufacturing contexts using the optimized path planning strategy

In the experiments of this study, we have got all three optimal planners (RRT, FMT, and T-RRT) are unidirectional, single query, and non-lazy algorithms. However, for other applications, the optimal planners may not be unidirectional, single query, or non-lazy algorithms. In addition, if we change the values of W_A and W_B of Eq. (7) in human–robot collaborative tasks, some certain features of different path planners would also be altered. Therefore, more complicated and general robot path planning applications will be investigated in future work to reveal deeper property comparisons of unidirectional and bidirectional, single query and multiple query, and non-lazy and lazy for these path planners in smart manufacturing contexts.

Acknowledgment

This work was supported by the National Science Foundation under Grant No.: IIS-1845779.

References

[1] Kang, H. S., Lee, J. Y., Choi, S., Kim, H., Park, J. H., Son, J. Y., Kim, B. H., and Noh, S. D., 2016, "Smart Manufacturing: Past Research, Present Findings, and Future Directions," Int. J. Precis. Eng. Manuf.-Green Technol., **3**(1), pp. 111–128.

[2] Westerlund, L., 2000, *The Extended Arm of Man: A History of Industrial Robot*, Informationsförlaget, Stockholm, Sweden.

[3] Wang, W., Li, R., Chen, Y., and Jia, Y., 2018, “Human Intention Prediction in Human-Robot Collaborative Tasks,” Proceedings of 2018 ACM/IEEE International Conference on Human-Robot Interaction, Chicago, IL, Mar. 5–8, pp. 279–280.

[4] Wang, W., Li, R., Diekel, Z. M., and Jia, Y., 2018, “Robot Action Planning by Online Optimization in Human-Robot Collaborative Tasks,” Int. J. Intell. Rob. Appl., **9**(23), pp. 1–19.

[5] Iqbal, J., Islam, R. U., Abbas, S. Z., Khan, A. A., and Ajwad, S. A., 2016, “Automating Industrial Tasks Through Mechatronic Systems—A Review of Robotics in Industrial Perspective,” Tehnički Vjesnik, **23**(3), pp. 917–924.

[6] Bahrin, M. A. K., Othman, M. F., Azli, N. H. N., and Talib, M. F., 2016, “Industry 4.0: A Review on Industrial Automation and Robotic,” Jurnal Teknologi, **78**(6–13), pp. 137–143.

[7] Fang, S., Huang, X., Chen, H., and Xi, N., 2016, “Dual-Arm Robot Assembly System for 3C Product Based on Vision Guidance,” 2016 IEEE International Conference on Robotics and Biomimetics (ROBIO), Qingdao, China, Dec. 3–7, pp. 807–812.

[8] Cheein, F. A., Herrera, D., Gimenez, J., Carelli, R., Torres-Torriti, M., Rosell-Polo, J. R., Escolà, A., and Arnó, J., 2015, “Human-Robot Interaction in Precision Agriculture: Sharing the Workspace With Service Units,” 2015 IEEE International Conference on Industrial Technology (ICIT), Seville, Spain, Mar. 17–19, pp. 289–295.

[9] Strabala, K. W., Lee, M. K., Dragan, A. D., Forlizzi, J. L., Srinivasa, S., Cakmak, M., and Micelli, V., 2013, “Towards Seamless Human-Robot Handovers,” J. Hum.-Rob. Interact., **2**(1), pp. 112–132.

[10] Cakmak, M., Srinivasa, S. S., Lee, M. K., Kiesler, S., and Forlizzi, J., 2011, “Using Spatial and Temporal Contrast for Fluent Robot-Human Hand-Overs,” Proceedings of the 6th International Conference on Human-Robot Interaction, Lausanne, Switzerland, Mar. 6–9, pp. 489–496.

[11] Kuffner, J. J., and LaValle, S. M., 2000, “RRT-connect: An Efficient Approach to Single-Query Path Planning,” Proceedings of ICRA’00 IEEE International Conference on Robotics and Automation, San Francisco, CA, Apr. 24–28, pp. 995–1001.

[12] Kamil, F., Tang, S., Khaksar, W., Zulkifli, N., and Ahmad, S., 2015, “A Review on Motion Planning and Obstacle Avoidance Approaches in Dynamic Environments,” Adv. Rob. Autom., **4**(2), pp. 134–142.

[13] Lozano-Pérez, T., and Wesley, M. A., 1979, “An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles,” Commun. ACM, **22**(10), pp. 560–570.

[14] Tang, S. H., Khaksar, W., Ismail, N., and Ariffin, M., 2012, “A Review on Robot Motion Planning Approaches,” Pertanika J. Sci. Technol., **20**(1), pp. 15–29.

[15] LaValle, S. M.,1998, Rapidly-Exploring Random Trees: A New Tool for Path Planning.

[16] Janson, L., Schmerling, E., Clark, A., and Pavone, M., 2015, “Fast Marching Tree: A Fast Marching Sampling-Based Method for Optimal Motion Planning in Many Dimensions,” Int. J. Rob. Res., **34**(7), pp. 883–921.

[17] Kavraki, L., Svestka, P., and Overmars, M. H., 1996, “Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces,” IEEE Trans. Robot. Automat., **12**(4), pp. 566–580.

[18] Karaman, S., and Frazzoli, E., 2011, “Sampling-Based Algorithms for Optimal Motion Planning,” Int. J. Rob. Res., **30**(7), pp. 846–894.

[19] Hsu, D., Latombe, J.-C., and Motwani, R., 1999, “Path Planning in Expansive Configuration Spaces,” Int. J. Comput. Geom. Appl., **9**(4n05), pp. 495–512.

[20] Dobson, A., and Bekris, K. E., 2014, “Sparse Roadmap Spanners for Asymptotically Near-Optimal Motion Planning,” Int. J. Rob. Res., **33**(1), pp. 18–47.

[21] Devaurs, D., Siméon, T., and Cortés, J., 2013, “Enhancing the Transition-Based RRT to Deal with Complex Cost Spaces,” 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, May 6–10, pp. 4120–4125.

[22] Elbanhawi, M., and Simic, M., 2014, “Sampling-Based Robot Motion Planning: A Review,” IEEE Access, **2**(1), pp. 56–77.

[23] Sánchez, G., and Latombe, J.-C., 2003, “A Single-Query Bi-Directional Probabilistic Roadmap Planner With Lazy Collision Checking,” *Robotics Research*, Springer, New York, pp. 403–417.

[24] Bohlin, R., and Kavraki, L. E., 2000, “Path Planning Using Lazy PRM,” Proceedings 2000 ICRA. Millennium Conference, IEEE International Conference on Robotics and Automation, Symposia Proceedings (Cat. No. 00CH37065), San Francisco, CA, Apr. 24–28, pp. 521–528.

[25] Şucan, I. A., and Kavraki, L. E., 2009, Kinodynamic Motion Planning by Interior-Exterior Cell Exploration,” *Algorithmic Foundation of Robotics* VIII, Springer, New York, pp. 449–464.

[26] Koppula, H. S., Gupta, R., and Saxena, A., 2013, “Learning Human Activities and Object Affordances From rgb-d Videos,” Int. J. Rob. Res., **32**(8), pp. 951–970.

[27] Wang, W., Li, R., Chen, Y., Diekel, Z., and Jia, Y., 2018, “Facilitating Human-Robot Collaborative Tasks by Teaching-Learning-Collaboration From Human Demonstrations,” IEEE Trans. Autom. Sci. Eng., **16**(2), pp. 1–14.

[28] Wang, W., Li, R., Diekel, Z. M., Chen, Y., Zhang, Z., and Jia, Y., 2019, “Controlling Object Hand-Over in Human–Robot Collaboration Via Natural Wearable Sensing,” IEEE Trans. Hum.-Mach. Syst., **49**(1), pp. 59–71.

[29] Wang, W., Li, R., Diekel, Z. M., and Jia, Y., 2018, “Hands-Free Maneuvers of Robotic Vehicles via Human Intentions Understanding Using Wearable Sensing,” J. Rob., **2018**(1), pp. 1–10.

[30] Rossi, M., Benatti, S., Farella, E., and Benini, L., 2015, “Hybrid EMG Classifier Based on HMM and SVM for Hand Gesture Recognition in Prosthetics,” 2015 IEEE International Conference on Industrial Technology (ICIT), Seville, Spain, Mar. 17–19, pp. 1700–1705.

[31] Gunter, S., and Bunke, H., 2003, “Optimizing the Number of States, Training Iterations and Gaussians in an HMM-Based Handwritten Word Recognizer,” Seventh International Conference on Document Analysis and Recognition, Edinburgh, Scotland, Aug. 3–6, pp. 472–476.

[32] Jain, A., Nandakumar, K., and Ross, A., 2005, “Score Normalization in Multimodal Biometric Systems,” Pattern Recognit., **38**(12), pp. 2270–2285.

[33] Wang, W., Liu, N., Li, R., Chen, Y., and Jia, Y., 2018, “HuCoM: A Model for Human Comfort Estimation in Personalized Human-Robot Collaboration,” ASME 2018 Dynamic Systems and Control Conference, Atlanta, GA, Sept. 30–Oct. 3, pp. 1–6.

[34] Sucan, I. A., Moll, M., and Kavraki, L. E., 2012, “The Open Motion Planning Library,” IEEE Rob. Autom. Mag., **19**(4), pp. 72–82.

[35] Chitta, S., Sucan, I., and Cousins, S., 2012, “Moveit!,” IEEE Rob. Autom. Mag., **19**(1), pp. 18–19.