Improved Benchmarking for Steering Algorithms

Mubbasir Kapadia[1,2], Matthew Wang[1], Glenn Reinman[1], and Petros Faloutsos[1,3]

[1]University of California, Los Angeles
[2]University of Pennsylvania
[3]York University

Abstract. The statistical analysis of multi-agent simulations requires a definitive set of benchmarks that represent the wide spectrum of challenging scenarios that agents encounter in dynamic environments, and a scoring method to objectively quantify the performance of a steering algorithm for a particular scenario. In this paper, we first recognize several limitations in prior evaluation methods. Next, we define a measure of *normalized effort* that penalizes deviation from desired speed, optimal paths, and collisions in a single metric. Finally, we propose a new set of benchmark categories that capture the different situations that agents encounter in dynamic environments and identify truly challenging scenarios for each category. We use our method to objectively evaluate and compare three state of the art steering approaches and one baseline reactive approach. Our proposed scoring mechanism can be used (a) to evaluate a single algorithm on a single scenario, (b) to compare the performance of an algorithm over different benchmarks, and (c) to compare different steering algorithms.

1 Introduction

Goal driven autonomous agents are used to populate dynamic virtual worlds in a wide variety of applications ranging from urban simulations, movies, games, and education. A large variety of approaches have been proposed to address the problems of steering and navigation in dynamic environments. However, evaluating the performance of steering techniques is still a fundamental open problem.

Crowd simulations are evaluated using one of the following methods: (a) manual inspection, (b) comparison to real-world data, or (c) statistical analysis. In this work, we focus on the use of computational methods and statistical tools to analyze, evaluate, and test crowd simulations. The statistical analysis of multi-agent simulations requires a definitive set of benchmarks that represent the wide spectrum of challenging scenarios that agents encounter in dynamic environments, and a scoring method to objectively quantify the performance of a steering algorithm for a particular scenario.

Prior work has proposed a rich set of application-specific benchmarks and metrics to evaluate and analyze crowd simulations. The test cases are usually

limited to a small set of manually designed test cases, and ad hoc, scenario-dependent criteria. The work of [7] uses *presence* as a metric for crowd evaluation. The number of collisions and a measure of effort are often used as quantities that need to be minimized by steering algorithms [2, 8]. The work in [3] uses the "rate of people exiting a room" to analyze evacuation simulations. Many other approaches simply rely on visual fidelity and a subjective evaluation of the simulation. The work in [4] provides users with the flexibility of defining derived metrics in order to specify and detect custom behaviors in multi-agent simulations.

SteerBench [10] is the first work to propose a comprehensive set of manually defined benchmarks and a scoring method to objectively compare different steering algorithms in an application independent manner. However, it suffers from limitations that need to be addressed in order to provide a standard method of evaluating current and future steering approaches. The work in [6] performs random sampling in the space of obstacle and agent configurations to generate a very large set of *representative scenarios* that represent all the possible configurations an agent is likely to encounter while steering in dynamic environments. In addition, it defines three metrics: (1) success, (2) normalized time, and (3) normalized path length in order to objectively evaluate steering algorithms. These metrics can be used to measure coverage, quality, and failure for steering algorithms in the representative scenario space.

In this paper, we first evaluate and analyze three state of the art steering algorithms and one baseline reactive approach using SteerBench. The three steering algorithms are: (a) a local field based approach that performs steering and implicit space-time planning [5], (2) a hybrid method that combines reaction, prediction, and planning for steering and navigation [9], and (3) a method based on reciprocal velocity obstacles for collision avoidance [1]. From our first hand experience with SteerBench, we identify important limitations and open questions that need to be addressed (Section 2). Second, we propose a measure of effort that can be used to effectively measure the performance of a steering algorithm for a particular scenario (Section 3). Our scoring measure penalizes sub-optimal paths, deviations from the desired speed of an agent, and collisions in a single metric without the need of arbitrarily combining metrics with different units. In addition, we propose a measure of *optimal effort*. By normalizing the score with respect to an optimal, our scores have meaning on their own. They can therefore be used to compare the performance of an algorithm across different scenarios, as well as to compare different algorithms on the same scenario. Third, we propose an improved set of benchmark categories, and procedurally generate a large number of scenarios for each category in order to identify a definitive set of *challenging scenarios* that agents encounter in dynamic environments (Section 4). Finally, Section 5 presents a rigorous evaluation of three steering algorithms using the improved benchmarks and metrics, and Section 6 concludes the paper.

2 Experience with SteerBench

SteerBench [10] provides a benchmark suite of 38 scenarios (56 test cases) which are used to challenge a steering algorithm in the following broad categories: (1) simple validation scenarios, (2) basic agent-agent interactions, (3) agents interacting in presence of obstacles, (4) group interactions, and (5) large scale scenarios. In addition, it proposes the following primary metrics to evaluate the efficiency of a steering algorithm: (1) number of collisions, (2) total time, and (3) total energy. A weighted sum of the three metrics is used to compute a *score* which can serve as a comparative measure for different steering algorithms. We use SteerBench to evaluate and compare four steering techniques: (1) `EGOCENTRIC` [5], (2) `PPR` [9], (3) `RVO` [1], and (4) one baseline reactive approach. The SteerBench scores of these four algorithms are provided in Table 1. The average time per agent in reaching goal (seconds) and total energy spent per agent ($kg \cdot m^2/s^2$) for `EGOCENTRIC` is also provided for reference. In this section, we describe our experience with using SteerBench and identify some limitations and open questions that need to be addressed. Please note that these limitations are not due to *bugs* but are fundamental shortcomings in the method of evaluation.

Observations. The four algorithms, including the reactive approach, can successfully solve 36 out of the 38 scenarios provided in SteerBench. We observe little or no difference in the scores of all four algorithms on the simple scenarios and basic agent interactions. The values of the scores range from 100 to 500 depending on the length and difficulty of the scenario. This only allows us to compare the performance of different steering algorithms on the same scenario and prevents the score from being considered on its own or across different scenarios.

The `Oncoming-groups` is a challenging scenario that results in different behaviors from the three approaches. We notice that `EGOCENTRIC` results in group formations where the the two oncoming groups of agents stick together and smoothly maneuver around the other group, taking a longer but collision-free trajectory. In `PPR` and `REACTIVE`, the agents do not deviate from their optimal trajectories and resort to reactively avoid oncoming threats. However, the resulting scores of these approaches is not much different and does not capture these emergent and vastly different behaviors.

The `Overtake` scenarios were designed to test the ability of an agent to pass an agent from behind while in a narrow passage. We observe that the scores for all 4 algorithms are approximately the same. However, However, visual inspection of the simulation shows that `PPR` and `REACTIVE` did not demonstrate an overtaking behavior. The `Surprise` scenarios challenge agents to suddenly react to crossing threats in narrow corridoors. However, the effect of the interesting interaction between agents on the overall score is diluted due to the length of the scenario. Finally, the scenarios with $3-4$ agents interacting with one another all have approximately the same score. Agent interactions can be vastly different depending on the initial conditions and manually designing a few scenarios to

test interactions between agents is not sufficient.

We identify the following limitations from our first hand experience with SteerBench:

- The evaluation of the steering algorithm is limited to the 56 hand designed test cases that are provided with SteerBench which cannot capture the entire spectrum of scenarios that an agent may encounter while steering in dynamic environments and are prone to the problem of overfitting.
- Often, the most interesting portion of a scenario is the interaction between agents and obstacles which is only a small portion of the scenario. The analysis of the entire scenario thus reduces the effect of the *interaction of interest* on the cumulative score.
- The metrics are scenario dependent, i.e., the scores produced by SteerBench vary greatly over different scenarios. This is because the metrics are computed in a time dependent manner. As a result, scenarios that take longer to complete have larger scores. As a result, it is only meaningful to compare the scores of two simulations for a test case. However, it is impossible to evaluate the efficiency of steering algorithms independently for one scenario over different scenarios.
- There exists no definition of ground truth or optimality for the scenarios. The notion of a perfect score for a particular scenario would provide a strong basis for comparison and help better identify the areas of a particular algorithm that needs improvement.
- The scores are only intended to serve as a basis for comparison between two algorithms and have no meaning on an absolute scale.
- The weights used to sum three primary metrics also did not have any intuitive meaning.

Hence, there exists no definitive set of benchmarks that represent the wide variety of challenging scenarios that agents encounter in complex virtual worlds. Also, we need metrics that can measure the performance of an algorithm in a time and scenario independent fashion. This greatly limits the objective evaluation and comparison of different steering and navigation techniques.

3 Metrics for Evaluation

In this section, we propose a bio-mechanically inspired measure of effort to objectively score the performance of a steering algorithm on a particular scenario. We also calculate the optimal value of effort required to solve a scenario which allows us to normalize our score. Our proposed scoring mechanism can be used (a) to evaluate a single algorithm on a single scenario, (b) to compare the performance of an algorithm across different benchmarks, and (c) to compare different steering algorithms.

The work in [12] describes that steering agents should obey two principles while navigating in dynamic environments: (1) they should minimize the distance

Test Case	Time	Energy	Egocentric	RVO	PPR	Reactive
Simple-3	5.75	112.0	117.76	114.41	118.77	117.87
Simple-obstacle-2	14.2	253.13	267.33	265.67	268.23	268.23
Curves	21.5	363.85	385.35	431.22	385.5	385.5
Crossing-6	22.2	277.8	298.32	298.94	295.25	295.25
Oncoming-obstacle	16.83	267.87	284.7	289.64(0.5)	284.1	276.75
Oncoming-groups	41.72	598.87	640.5	643.83	637.53	638.75
Fan-out	32.25	519.98	552.24	549.48	551.3	551.3
Cut-across-2	33.57	505.9	539.49	546.82	537.1	536.9
Surprise-2	25.7	353.84	429.54(1)	401.56	407.15	406.62
Overtake-obstacle	16.9	279.27	296.17	300.28	291.23	297.6
4-way-confusion-2	15.26	253.4	268.67	267.0	269.13	265.63
Double-squeeze	22.63	308.43	331.05	371.3(1)	354.6(0.5)	379.5(1)
Doorway-two-way	–	–	Fail	331.38	Fail	Fail
Wall-squeeze	–	–	Fail	Fail	434.43(2)	Fail

Table 1: Evaluation Results using SteerBench – Lower score is better. Numbers in () is the average number of collisions per agent.

traveled in reaching their destination, and (2) they should attempt to move at their preferred speed. A collision-free trajectory is a fundamental requirement that must also be met. A simple effort function that measures the distance travelled by an agent to reach the goal does not address the influence of speed. Similarly, a metric that only measures the time to reach the goal will result in the agents walking at their maximum speed rather than at their preferred speed, expending more energy than necessary.

The Principle of Least Effort states that an organism will maintain on average the least possible work expenditure rate as estimated by itself. When applied to steering, it means that agents will naturally choose a path to their goal which they expect will require the least amount of *effort*. Biomechanics research has quantified the energy expended by a walking human as a function of the subject's instantaneous velocity [13]. The effort, $E_m^a(s)$ of an agent a as the metabolic energy expended while walking along a path for a given scenario s is computed as follows:

$$E_m^a(s) = m \int_{t=0}^{t=T} e_s + e_w |\mathbf{v}|^2 dt. \tag{1}$$

Here, m is the mass of an agent, T is the total time of the simulation, and e_s, e_w are per agent constants. For an average human, $e_s = 2.23 \frac{J}{Kg \cdot s}$ and $e_w = 1.26 \frac{J \cdot s}{Kg \cdot m^2}$.

Collision Effort. We introduce an effort penalty, $E_c^a(s)$ for collisions. For every second that an agent, a is in a state of collision, this penalty is a function of the penetration of the collision.

$$E_c^a(s) = m \int_{t=0}^{t=T} e_c c_p(t) dt, \tag{2}$$

where $e_c = 10\frac{J}{Kg \cdot m \cdot s}$ is a penalty constant for collisions. The collision penetration function, $c_p(t)$ estimates the current penetration depth of the collision if the agent is colliding with another agent at that point of time.

Optimal Effort. The optimal effort for an agent a in a scenario s is defined as the energy consumed in taking the optimal route to the target while traveling at the average walking speed $= \sqrt{\frac{e_s}{e_w}} = 1.33 m/s$. Let L_{opt}be the optimal length for an agent a to reach the target. The optimal effort, $E_{opt}^a(s)$ for an agent a is calculated as follows:

$$E_{opt}^a(s) = 2mL_{opt}\sqrt{e_s e_w}. \tag{3}$$

The derivation of Equation 3 can be found here [2]. We calculate L_{opt} as the length along the optimal trajectory (found using A*) for an agent to reach its goal, taking into account only static obstacles.

Normalized Effort. The normalized effort for a particular agent a in a scenario s is defined as the ratio of the optimal effort in reaching a target to the actual effort taken, accounting for collisions. It is calculated as follows:

$$E_r^a(s) = \frac{E_{opt}^a(s)}{E_m^a(s) + E_c^a(s)} \tag{4}$$

The normalized effort for all agents for a given scenario is calculated as follows:

$$E_r(s) = \frac{\sum_{a=1}^{a=N} E_r^a(s)}{N} \tag{5}$$

where N is the number of agents in the scenario. The value of $E_r(s)$ ranges from 0 to 1 with a higher value indicating better performance for a given scenario. A *perfect* steering algorithm would have $E_r(s) = 1$.

Average Quality. The average quality of a steering algorithm over a set of scenarios is computed as the average value of $E_r(s)$ for all scenarios.

4 Benchmarks for Evaluation

Based on the work in [6], we define a scenario as one possible configuration of agents and obstacles. A large number of scenarios can be generated by randomly sampling agent and obstacle configurations. However, in the majority of cases, it

is of particular interest to define scenarios which capture challenging interactions between agents. Trivial scenarios where agents need not perform any steering to reach their destination (i.e. agents never interact with one another) are generally not going to provide a meaningful comparison. To ensure agent interactions, we place a constraint on scenario generation such that all agents must interact (i.e. their optimal paths must cross in space and time). The resulting scenarios generated focus on more *interesting interactions* between agents, and therefore avoid diluting evaulation scores in our methodology by measuring trivial steering simulations where agents do not interact. We also provide the flexibility place additional user-defined constraints on the generated scenario in order to meet certain criteria in order to define specific categories of scenarios.

We define a set of benchmark categories that uniquely capture the different *challenges* that steering and navigating agents encounter in dynamic worlds. These benchmark categories are described below.

- **Single Agent Navigation.** These scenarios have one agent with a fixed initial and desired position. We randomly sample obstacle configurations in order to evaluate the navigation capabilities of an algorithm. Figure 1(a) illustrates an example scenario generated for this category.
- **Agent Interactions.** These scenarios represent different configurations of oncoming as well crossing agents (Figure 1(b)-(c)). Agents are randomly positioned at the boundary with goals at the opposite end of the environment to ensure that all agents will arrive at the center of the environment at approximately the same time, thus forcing an interaction. An obstacle is randomly positioned in the center to pose an additional challenge. The number of agents is varied from $2 - 10$.
- **Narrow Passages.** These scenarios challenge oncoming agents to travel in narrow passages that are just big enough to allow two agents to pass through (Figure 1(d)). The number of agents is varied from $2 - 4$.
- **Narrow Crossings.** These scenarios capture combinations of oncoming and crossing interactions between agents in narrow corridoors (Figure 1(e)-(f)).
- **Oncoming Groups.** The scenarios in this category represent interactions between oncoming groups of agents (Figure 1(g)-(h)). Agents are randomly positioned on two opposing sides of the environment forming two oncoming groups. Different obstacle configurations may also be randomly positioned in the center of the environment to pose as an additional challenge. The number of agents in the group is varied from $2 - 5$.
- **Crossing Groups.** These scenarios represent interactions between crossing groups of agents (Figure 1(i)-(j)). Agents are randomly positioned in 2 adjacent groups which interact in the center of the environment. Obstacles may also be randomly positioned in the center of the environment. The number of agents in each group is varied from $2 - 5$.
- **Group Confusion.** This category captures interactions between 4 groups of agents that arrive at the center of the environment from opposite sides (Figure 1(k)-(l)). The number of agents in each group is varied from $2 - 4$.

We randomly generate $10,000$ scenario samples for each of the benchmark categories and and calculate the mean of the average quality of the three steering algorithms [1, 5, 9]. Figure 2 illustrates the average quality distribution for the benchmark categories described above. We identify the 100 scenarios with the lowest quality scores as the failure set for each benchmark category. These are highlighted in blue in Figure 2. The next 900 scenarios (highlighted in red) with lowest quality measures are identified as challenging scenarios for the respective benchmark category. The average quality thresholds for the failure set and the challenging scenarios are given in Table 2.

Benchmark Category	Failure Threshold	Challenge Threshold
Single Agent Navigation	0.125	0.481
Agent Interactions	0.212	0.303
Agent Interactions Obstacle	0.136	0.225
Narrow Passages	0.322	0.566
Narrow Crossings	0.245	0.631
Oncoming Groups	0.192	0.322
Oncoming Groups Obstacle	0.115	0.176
Crossing Groups	0.222	0.380
Crossing Groups Obstacle	0.149	0.258
Group Confusion	0.112	0.177
Group Confusion Obstacle	0.132	0.175

Table 2: The average quality thresholds used to identify the failure set and the challenging scenarios for each benchmark category. In Figure 2, the scenarios which fall below are the failure threshold are highlighted in blue while the challenging scenarios are highlighted in red.

5 Results

In this section, we evaluate `EGOCENTRIC`,`PPR`,`RVO` and `REACTIVE` using the proposed metrics and benchmark categories described in Section 3 and Section 4. Table 3 provides the average quality measures of the four steering algorithms for the aforementioned benchmark categories. The `FAIL` quality measure describes the quality of the algorithm for the scenarios in the failure set. The `CHALLENGE` quality measures describes the quality measure of the algorithm on the challenging scenarios. Finally, the `ALL` quality measure describes the quality of the algorithm on the remaining sampled scenarios.

Observations. The `Single Agent Navigation` benchmarks are primarily used to test the planning abilities of the algorithms. We observe that the three standard algorithms have similar quality measures while `REACTIVE` performs particularly poorly as it constantly steers towards a local target that is chosen by casting

a ray towards the goal. In contrast, the quality measures of REACTIVE are comparable to PPR for the benchmarks involving agent interactions, as these scenarios challenge the reactive behavior of agents to avoid other dynamic threats. The Narrow Passages and Narrow Crossing are particularly challenging benchmarks as it challenges the ability of the steering agents to predictively avoid oncoming and crossing threats and prevent possible deadlock situations. A large percentage of the energy calculation in these two benchmark categories was due to collisions where agents were simply unable to predictively avoid oncoming and crossing threats and simply resorted to colliding with other agents. For all the Group Interactions benchmarks, we observe that PPR and REACTIVE both have similar quality measures. This is because PPR turns off predictions in the presence of crowds (more than 4 agents) and resorts to purely reactive behavior which is reflected in the scores. EGOCENTRIC outperforms the other algorithms in the group interactions due to the emergence of group behavior where nearby located agents tend to stick together while handling other agent groups.

Benchmark Category	Quality	Egocentric	RVO	PPR	Reactive
Single	FAIL	0.102	0.154	0.113	0.0076
Agent	CHALLENGE	0.321	0.385	0.334	0.121
Navigation	ALL	0.691	0.776	0.715	0.543
	FAIL	0.204	0.210	0.203	0.198
Agent	CHALLENGE	0.271	0.267	0.260	0.257
Interactions	ALL	0.653	0.643	0.641	0.645
Agent	FAIL	0.132	0.142	0.121	0.113
Interactions	CHALLENGE	0.191	0.193	0.183	0.178
Obstacle	ALL	0.601	0.613	0.564	0.546
	FAIL	0.302	0.256	0.278	0.132
Narrow	CHALLENGE	0.476	0.432	0.452	0.332
Passages	ALL	0.744	0.732	0.734	0.687
	FAIL	0.182	0.174	0.178	0.123
Narrow	CHALLENGE	0.465	0.452	0.445	0.343
Crossings	ALL	0.781	0.755	0.742	0.698
	FAIL	0.182	0.171	0.161	0.157
Oncoming	CHALLENGE	0.312	0.286	0.255	0.253
Groups	ALL	0.656	0.617	0.572	0.567
Oncoming	FAIL	0.133	0.127	0.103	0.097
Groups	CHALLENGE	0.189	0.165	0.145	0.138
Obstacle	ALL	0.465	0.426	0.392	0.390
	FAIL	0.221	0.209	0.193	0.192
Crossing	CHALLENGE	0.348	0.327	0.305	0.301
Groups	ALL	0.667	0.643	0.603	0.612
Crossing	FAIL	0.156	0.143	0.134	0.134
Groups	CHALLENGE	0.245	0.223	0.204	0.210
Obstacle	ALL	0.534	0.509	0.481	0.491
	FAIL	0.125	0.123	0.101	0.091
Group	CHALLENGE	0.167	0.156	0.145	0.143
Confusion	ALL	0.512	0.476	0.428	0.412
Group	FAIL	0.142	0.134	0.121	0.115
Confusion	CHALLENGE	0.175	0.167	0.154	0.144
Obstacle	ALL	0.412	0.387	0.341	0.324

Table 3: Average Quality of EGOCENTRIC,RVO,PPR, and REACTIVE on all benchmark categories: (1) FAIL: Failure Set (100 most difficult scenarios generated for that category). (2) CHALLENGE: Challenging Scenarios (1000 most difficult scenarios generated for that category, excluding the failure set). (3) ALL: The remaining 9000 scenarios that were generated.

6 Conclusion

In this paper, we propose a set of benchmark categories to capture different situations that steering agents encounter in dynamic environments and a measure of *normalized effort* that penalizes deviation from desired speed, optimal paths, and collisions in a single metric. We use our method to objectively evaluate and compare three state of the art steering approaches and one baseline reactive approach. Our proposed scoring mechanism can be analyzed on its own, can be used to compare the performance of an algorithm over different benchmarks, and also be used to compare different steering algorithms. For future work, we would like to analyse the performance of steering approaches based on principles of energy minimization [2] and approaches that use more complex locomotion models [11]. We would also like to to compute metrics for real crowds to serve as *ground truth* for the benchmarks.

Acknowledgements

Intel supported this research through a visual-computing grant and the donation of a 32-core Emerald Ridge system with Xeon X7560 processors. We thank Randi Rost and Scott Buck from Intel for their support.

References

1. van den Berg, J., Lin, M.C., Manocha, D.: Reciprocal velocity obstacles for real-time multi-agent navigation. In: Proceedings of ICRA. pp. 1928–1935. IEEE (2008)
2. Guy, S.J., Chhugani, J., Curtis, S., Dubey, P., Lin, M., Manocha, D.: Pledestrians: a least-effort approach to crowd simulation. pp. 119–128. SCA (2010)
3. Helbing, D., Farkas, I., Vicsek, T.: Simulating dynamical features of escape panic. NATURE 407, 487 (2000)
4. Kapadia, M., Singh, S., Allen, B., Reinman, G., Faloutsos, P.: Steerbug: an interactive framework for specifying and detecting steering behaviors. In: SCA '09. pp. 209–216. ACM (2009)
5. Kapadia, M., Singh, S., Hewlett, W., Faloutsos, P.: Egocentric affordance fields in pedestrian steering. In: I3D '09. pp. 215–223. ACM (2009)
6. Kapadia, M., Wang, M., Singh, S., Reinman, G., Faloutsos, P.: Scenario space: Characterizing coverage, quality, and failure of steering algorithms. In: SCA (2011)
7. Pelechano, N., Stocker, C., Allbeck, J., Badler, N.: Being a part of the crowd: towards validating vr crowds using presence. pp. 136–142. AAMAS (2008)
8. Shao, W., Terzopoulos, D.: Autonomous pedestrians. In: SCA. ACM (2005)
9. Singh, S., Kapadia, M., Hewlett, W., Faloutsos, P.: A modular framework for adaptive agent-based steering. In: I3D 2011. ACM (2011)
10. Singh, S., Kapadia, M., Naik, M., Reinman, G., Faloutsos, P.: SteerBench: A Steering Framework for Evaluating Steering Behaviors. CAVW (2009)
11. Singh, S., Kapadia, M., Reinman, G., Faloutsos, P.: Footstep navigation for dynamic crowds. CAVW (2011)
12. Still, K.G.: Crowd Dynamics. Ph.D. thesis, United Kingdon (2000)
13. Whittle, M.: Gait analysis: An introduction (1996)

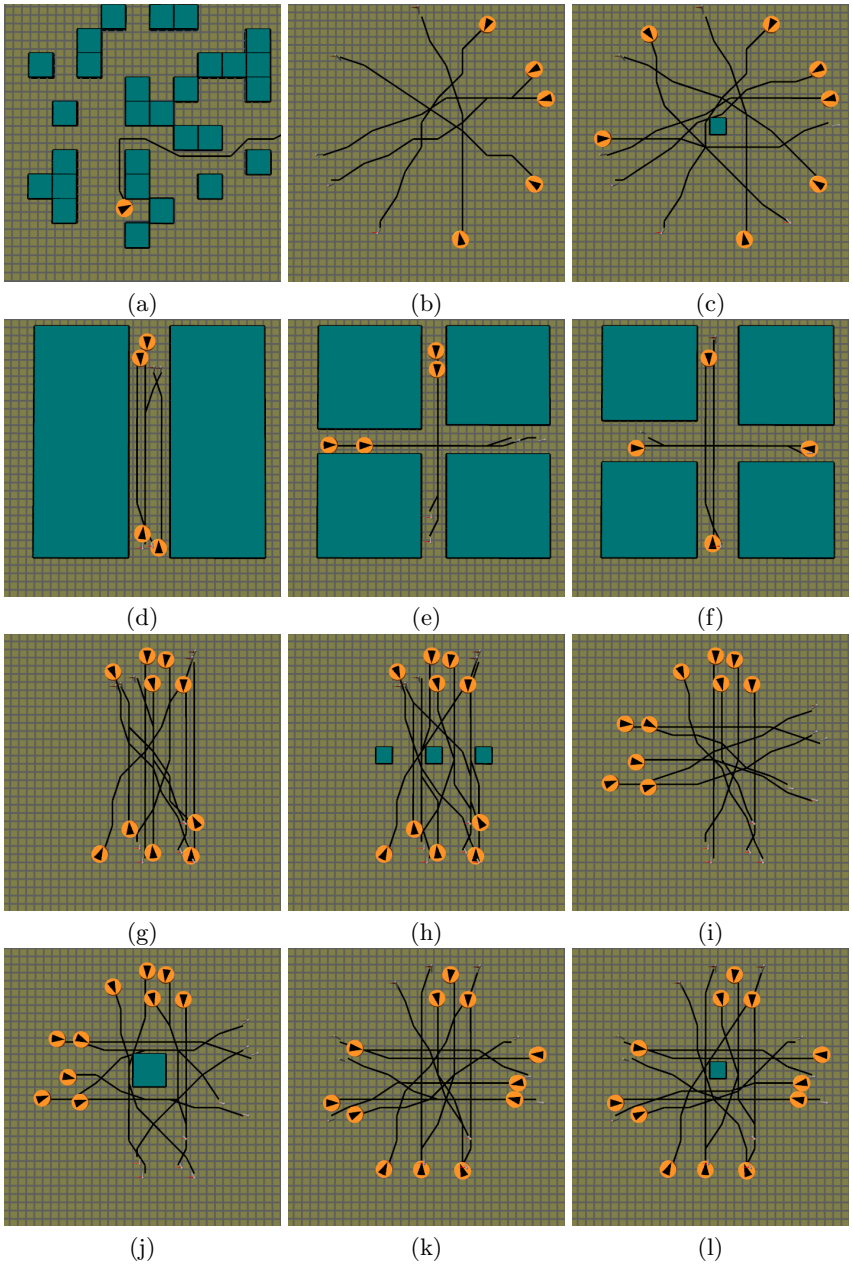

Fig. 1: Randomly generated scenarios for each of the benchmark categories.

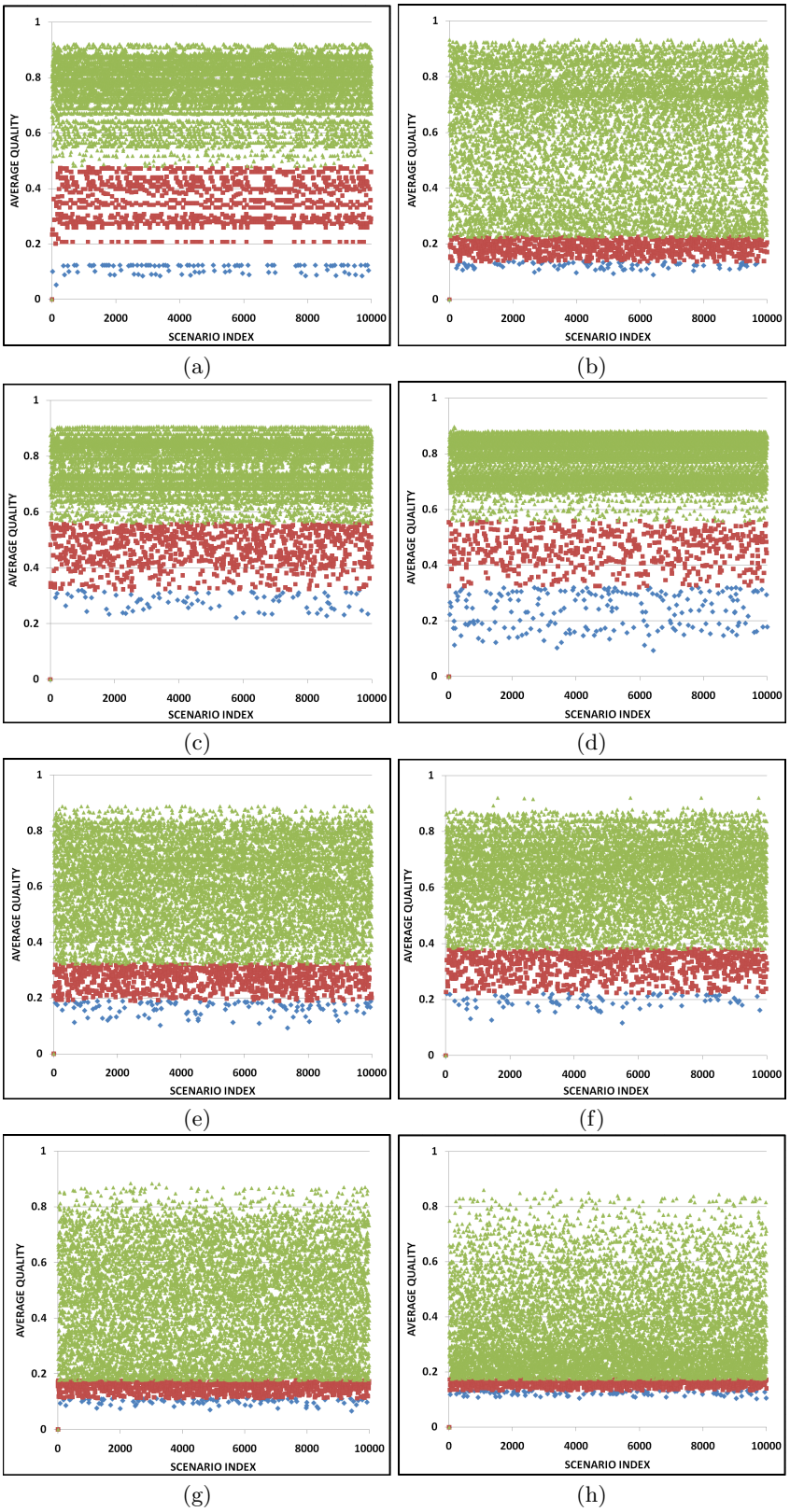

Fig. 2: Mean of average quality of `EGOCENTRIC,PPR,RVO` for the following benchmark categories. (a) Single agent navigation. (b) Agent interactions with obstacles. (c) Narrow passages. (d) Narrow crossings. (e) Oncoming groups. (f) Crossing groups. (g) Group confusion. (h) Group confusion with obstacles. The blue, red, and green points highlight the failure set, challenging scenarios, and the remaining scenarios respectively.