How to Best Automate Intersection Management

Aashiq Parker
Department of Computer Science
University of Cape Town
Cape Town, South Africa
PRKAAS003@myuct.ac.za

Geoff Nitschke
Department of Computer Science
University of Cape Town
Cape Town, South Africa
gnitschke@cs.uct.ac.za

***Abstract*—Recently there has been increased research interest in developing adaptive control systems for autonomous vehicles. This study presents a comparative evaluation of two distinct approaches to automated intersection management for a multi-agent system of autonomous vehicles. The first is a centralized heuristic control approach using an extension of the *Autonomous Intersection Management* (AIM) system. The second is a decentralized neuro-evolution approach that adapts vehicle controllers so as they collectively navigate intersections. This study tests both approaches for controlling groups of autonomous vehicles on a network of interconnected intersections, without the constraints of traffic lights or stop signals. These task environments thus simulate potential future scenarios where vehicles must drive autonomously without specific road infrastructure constraints. The capability of each approach to appropriately handle various types of interconnected intersections, while maintaining an efficient throughput of vehicles and minimizing delay is tested. Results indicate that neuro-evolution is an effective method for automating collective driving behaviors that are robust across a broad range of road networks, where evolved controllers yield comparable task performance or out-perform an AIM controller.**

I. Introduction

Recently there has been increasing research attention on producing adaptive control systems for autonomous vehicles. To accommodate such autonomous vehicles there have been proposals that current road and highway infrastructure undergo significant changes. For example, replacing traffic lights and stop signs and allowing autonomous vehicles to coordinate their own interactions so as to avoid collisions and safely navigate through intersections [1].

In the context of *Intelligent Transportation Systems*, there is a focus on sensor and communication capabilities to run efficient centralized intersection control policies. For example, Dresner and Stone [2] proposed a new automated intersection management system called *Autonomous Intersection Management* (AIM) for autonomous vehicles. AIM used a *First Come, First Served* (FCFS) policy for directing vehicles through intersections. Intersection management simulations demonstrated AIM as out-performing current forms of intersection control including traffic lights and stop signs, in terms of increased traffic throughput and decreased delays. The efficacy of AIM and the FCFS protocol was further validated by Fajardo et al. [3] and Fok et al. [4] presenting an experimental comparison of multiple autonomous vehicles using FCFS versus those using traditional traffic signals at an intersection. Results indicated AIM using FCFS yielded a significantly higher task performance in terms of reducing vehicle delay for a range traffic flow intensity at intersections.

In subsequent work, Quinlan et al. [5] demonstrated that AIM and the FCFS protocol was applicable to a real autonomous vehicle tasked with efficiently passing through intersections. The authors validated their approach via simulating other vehicles that were concurrently trying to pass through the intersection in real time.

Hausknecht et al. [6] extended AIM with a multi-agent system of vehicles that autonomously navigated networks of interconnected intersections. The authors examined various control policies such that vehicles dynamically altered their planned paths. Results indicated dynamically reversing the flow of traffic along certain lanes in a network of roads enabled efficient traffic throughput given changing traffic conditions.

In related work, Au et al. [7] introduced a modification of the AIM protocol to handle unbalanced traffic in networks of intersections, where road conditions were conducive to traffic jams. The modified protocol guaranteed that all vehicles eventually get a reservation. Results indicated that the new intersection control policy outperformed FCFS in unbalanced traffic, that is where there was a large discrepancy of traffic volume among the incoming roads.

Au et al. [8] also modified AIM such that it accounted for mechanical failures of vehicles and potential collisions within an intersection. The authors argued that preemptively computed evasion plans are essential for collision avoidance, as dynamic evasive actions are not always successful without pre-allocation of spaces in an intersection. However the modified version of AIM only considered a specific set of mechanical failures and thus collision avoidance plans. Au et al. [9] introduced another AIM modification that generated *set-point schedules* such that vehicles arrived at given positions at the correct time and with the correct velocity in an intersection. The authors demonstrated their scheduling algorithm as outperforming related heuristic-based schedulers that did not provide guaranteed vehicle arrival times and velocities.

A key limitation of AIM, the FCFS protocol, and its various extensions, is that these intersection management systems assume perfect traffic flow conditions and vehicle sensory information. For example, the *set-point scheduler* [9] was unable to handle uncertainty in traffic flow that causes erroneous arrival times and vehicle velocity. A notable exception is the work of Dresner and Stone [10], where AIM and the FCFS protocol are used in company with traffic lights in order to accommodate both human drivers and autonomous vehicles. In this case AIM uses traffic signals as an indicator for the autonomous vehicles, for example, assuming that human driven cars could enter the intersection at any time if their signal is green. Similarly, semi-

978-1-5090-4601-0/17/$31.00

autonomous vehicles were considered, allowing human drivers to potentially handle unpredictable traffic conditions [11].

Thus, AIM does not generally account for uncertain and unpredictable traffic conditions or dynamic obstacles, such as pedestrians with some random behavior. Such elements of unpredictable behavior, incomplete information and noisy sensory environments must be appropriately handled if autonomous vehicles are to be successfully utilized on physical roads and highways.

Another approach to *automated intersection management* [3], [12] that potentially handles such problems is to use completely decentralized control where each vehicle's controller automatically adapts as vehicles interact with their environment. That is, to automate the synthesis of vehicle controllers such that when vehicles interact a desired collective behavior emerges for any given environment. In the case of intersection management, the collective behavior is coordinated driving that emerges for a given road environment.

Neuro-Evolution (NE) is the automated adaptation of *Artificial Neural Networks* (ANNs) with evolutionary algorithms [13]. NE has been applied to evolve controllers in simulated land-based vehicles that must accomplish various tasks including road navigation [14], [15], automobile crash warning [16] and formation driving (platooning) [17]. However, there has been relatively little research on the evolution of collective driving behaviors for automating coordinated movement and maximizing traffic flow throughput on any given road environment. NE was selected as it has demonstrated broad applicability to controller evolution in related collective behavior applications such as unmanned aerial vehicles [18], [19], [20] and aquatic robots [21], [22].

This study uses NE to automate the synthesis of collective driving behaviors for a given set of road networks (interconnected intersections), where there are no stop signals or traffic lights to assist with vehicle coordination and navigation. Rather, NE automates controller design where collective driving behavior emerges in response to the task of maximizing traffic throughput and minimizing delays at intersections. The research objectives of this study are two-fold.

A. Research Objectives

1) Demonstrate the efficacy of NE for collective driving behavior synthesis, where task performance is *average vehicle throughput* and *speed* on road networks of interconnected intersections.

2) Evaluate the efficacy of NE versus a centralized heuristic controller for autonomous intersection management, on the same road networks, given the same task performance metrics.

Objective one was derived given the current lack of research in the evolutionary design of collective driving behaviors in autonomous vehicle simulations that do not account for task constraints such as traffic lights or stop signals at road intersections. Objective two was derived given the significant amount of research on centralized heuristic-based intersection management systems for autonomous vehicles (for example, [2], [3], [4]), but the lack of evolutionary approaches for automating intersection management driving behavior.

TABLE I. NEURO-EVOLUTION PARAMETERS

Mutation rate	Add neuron	0.001
	Add connection	0.01
	Remove connection	0.001
	Weight	0.888
Survival rate		0.3
Disjoint excess gene recombination		0.1
Activation function		Sigmoid
Activation Scheme		Acyclic
Selection Operator		Roulette Wheel
Maximum species		5
Crossover proportion		0.4
Elitism proportion		0.1
Population Size		150
Generations		150
Input / Output Nodes		10 / 1
Complexity Threshold		33
Complexity Regulation		Absolute

TABLE II. EXPERIMENT AND SIMULATION PARAMETERS

Simulation runs	20
Task trials per generation	3
Task trial duration	90 seconds
Vehicles	48
AIM Parameters	
Protocol	FCFS
Trial Duration	90 seconds
Vehicle Speed	8
Request Cool-down	1.5 seconds
Task Performance Metrics	
Traffic throughput	Vehicle transit between road entry / exit points
Speed	Speed per simulation iteration

II. METHODS

This study evaluates a direct-encoding NE method for automating controller design in order that a multi-agent system of vehicles elicits collective driving behaviors appropriate for coordinating and navigating a road network of interconnected intersections. The controller evolution method tested was *Neuro-Evolution of Augmenting Topologies* (NEAT) [23]. Groups of vehicles were behaviorally and morphologically homogenous in that one evolved ANN controller and one sensor configuration was used by all vehicles. As a benchmark for the intersection management task, a modified version of AIM and the FCFS protocol was comparatively tested and evaluated on the same road networks.

A. AIM: Automated Intersection Management

The implementation of AIM uses the FCFS protocol for automated intersection management [2]. Vehicles followed preplanned routes through intersections, where vehicles continuously circled a given road network (figures 4 and 5) and average vehicle throughput and speed was calculated. When the vehicles reached an intersection they communicated with the intersection manager to simulate a path and produce a list of time-position tuples. This information was bundled into a request and sent to the intersection manager to reserve a path through the intersection.

The intersection manager stored a list of all reserved position-time tuples and used this to check incoming and future requests for collisions. Vehicles not approved to enter the intersection had to wait and then retransmit their request every 2.0 seconds. Using the FCFS policy, vehicles transmitting requests first were allowed to enter the intersection first, given no potential collisions. Also, a set of heuristics were added as an extension to the AIM simulator to prevent cases where lanes would overflow[1]. Table II presents the AIM simulation parameters used in this study.

B. NEAT for Controller Evolution

NEAT [23] is a direct encoding NE method[2] that has been demonstrated in previous research as appropriate for collective behavior evolution [24]. NEAT evolves both connection weights and ANN topologies, and applies three key techniques to maintain a balance between task performance and diversity of solutions. First, it assigns a unique historical marking to every new gene so as crossover can only be performed between pairs of matching genes. Second, NEAT speciates the population so as ANNs (genotypes) compete primarily within their own niches (identified by historical markings) instead of competing with the whole population. Third, NEAT begins evolution with a population of simple ANNs with no hidden nodes but gradually adds new topological structure (nodes and connections) using two special mutation operators, *add hidden node* and *add link*. A key advantage of NEAT is that this complexifying process is likely to find a solution in lower dimension search spaces compared to relatively large search spaces corresponding to large fixed topology ANNs specified *a priori*. This complexification process also makes NEAT amenable for solving a broad range of problems. Thus, NEAT is an established NE method that was selected as it has been successfully applied for controller evolution in similar tasks [25], [26], [27], [28], [29], [17].

1) NEAT Vehicle Controller: Vehicles used a feed-forward ANN comprised of 10 sensory input nodes connected to one motor output node (figure 1, left), where nine sensors were positioned at $22.5°$ intervals about the front periphery of the vehicle and one sensor was positioned at the rear. Sensors emulate *Infrared* (IR) proximity sensors in that the closest obstacle within a sensor's *Field of View* (FOV) returned the highest value. Sensor and motor output values were normalized in the range: $[0.0, 1.0]$. A sensor reading of 0.0 indicated no obstacles in the FOV and a reading of 1.0 indicated an obstacle at a given minimum distance in the sensor's FOV. A motor output value of 0.0 indicated the vehicle had stopped and a value of 1.0 indicated it was traveling at maximum speed.

NEAT evolved the number of hidden-layer nodes, connectivity and associated connection weight values between inputs, hidden and the output node. Thus, controller inputs and outputs were not subject to adaptation. An example (fittest) NEAT evolved controller is presented in figure 1 (center). Table I presents the NEAT parameters used in this study.

III. Experiments

Experiments tested 48 autonomous vehicles in a three dimensional physically realistic simulation of traffic passing through road networks of interconnected intersections. The simulator was custom designed using the *Unity* game engine[3] and modified to accommodate NEAT (section II-B) for controller evolution as well as AIM (section II-A) for automated intersection management.

Each method for automated intersection management was evaluated on 10 road networks (figures 4 and 5). In each experimental evaluation, 48 vehicles were initialized such that an equal number began at each of the start points on a road network. Each vehicle was assigned a random path to follow between a start and end point, where the number of start and end points was determined by the given road network. For example, in the road network illustrated in figure 1 (right) there are eight possible start and end points. All road networks were created using *Blender*[4] and imported into the simulator.

The 10 road networks were modeled after real traffic intersections in a major metropolitan area. Multiple instances of the intersections were linked with interconnecting roads such that vehicles must pass through multiple intersections in order to transit between entry and exit points on a given road network. Each road network included a set of way-points that vehicles followed to move between entry and exit points, where paths between pairs of way-points were defined using Bezier curves (figure 1, right).

Methods (NEAT and AIM) were evaluated on increasingly difficult road networks (figures 4 and 5). Task difficulty was equated with the number of start and end points, the number of lanes per road and hence the number of vehicles that could concurrently enter an intersection. Also, the narrowing of roads requiring traffic to merge, one way roads limiting the direction of traffic flow and the number of roads creating an intersection were all indicative of increased task difficulty. For each road network, the task was to automate the coordination of N vehicles, each following their own preset path through a road network (section I-A). Experiments thus measured the efficacy of NEAT and AIM (section II) for automating intersection management such that *traffic throughput* and *vehicle speed* was maximized (table II), and traffic delays were thus minimized.

A. AIM Method: Heuristic Control

In the case of AIM (section II-A), each experiment consisted of 20 simulation runs, where each run lasted 90 seconds. Each simulation time-step equalled $1/15th$ of a second where a vehicle could move up to a distance of ≈ 0.54 meters per time-step. The total distance to travel depended upon the specific road network tested.

In each AIM experiment an equal number of vehicles started at each entry point and a path between that entry point and a randomly selected exit point was calculated. However, to emulate traffic intermittently entering a road network, at each entry point to the road network, there was a random delay of $(0, 2]$ seconds between the spawning and movement of each vehicle. Given this degree of randomness introduced by variable start times and paths, 20 simulation runs were executed and an average task performance for each metric calculated (table II).

[1]Vehicle heuristics are available online: https://github.com/Amposter/Unity-AIM/blob/master/Appendices/Heuristics/Heuristics.md

[2]This study used SharpNEAT: http://sharpneat.sourceforge.net/

[3]https://unity3d.com/

[4]https://www.blender.org/

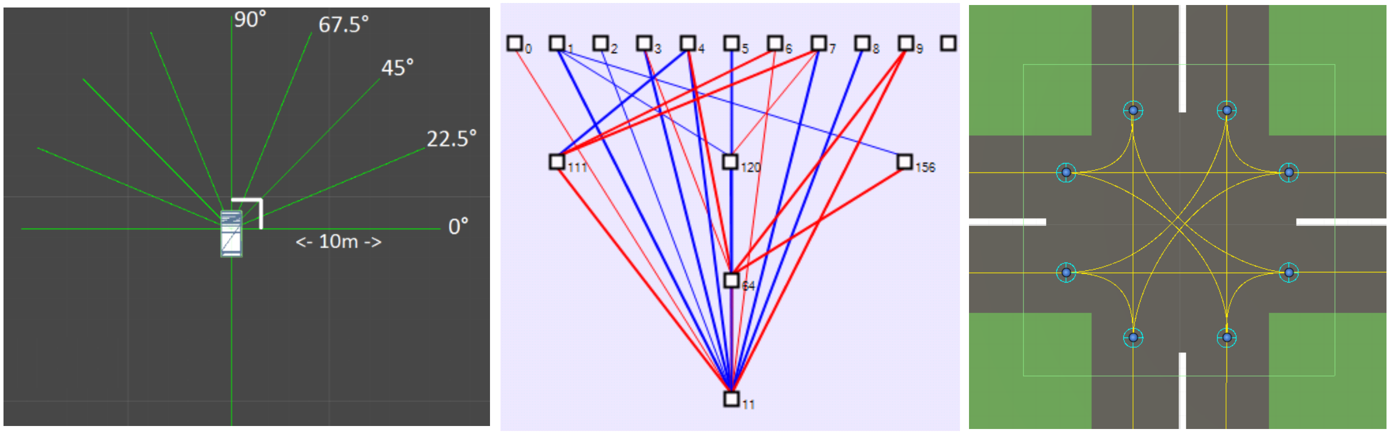

Fig. 1. *Left:* Vehicle sensory configuration uses 10 proximity sensors, with nine sensors covering a 180 degree field of view at the front of the vehicle and one sensor at the rear. *Center:* Fittest NEAT evolved controller for road network 10. The additional unlabeled input is a bias node. Blue and red connections are positive and negative weights respectively, where line thickness represents connection strength. *Right:* In the AIM simulator, each vehicle path is defined a set of nodes which defines a set of Bezier Curves used to calculate way points.

B. Neuro-Evolution (NE) for Controller Adaptation

Each NE experiment applied NEAT (section II-B) to evolve collective driving behavior such that all vehicles effectively navigated a road network of intersections, minimizing collisions and maximizing vehicle throughput. Each NE experiment evolved collective driving behavior for 150 generations (table I) on each of the 10 road networks (figures 4 and 5). These road networks were selected since they encapsulate many features common to real road networks. For example, multi-lane dual carriage-way roads, intersections with many points of entry and exit and one-way roads. One generation comprised three simulation task trials, where each task trial was 90 seconds (simulation iterations). Each task trial initialized an equal number of vehicles at each of the road network entry points, where vehicles began to move one at a time at $(0, 2]$ (randomly selected) second intervals.

Table I specifies all parameters and values used in the NE experiments. These values were derived in exploratory parameter tuning experiments, where similar values were found to yield comparable results. The vehicle group was behaviorally homogenous in that all vehicles used the same ANN controller. For each evolutionary run, the fittest controller was selected after 150 generations and an average fitness calculated over 20 runs. For each road network, the fittest evolved controller was executed in 20 evaluation runs (with no controller adaptation) on the same road network (section III-C).

1) Fitness Function: Equation 1 specifies the fitness function used to direct NEAT controller evolution.

$$f = (S * 2000) + (D * 1000) - (C * 250) - (B * 100) \quad (1)$$

Where: S is a value in the range: $[0, 1]$ representing average vehicle speed, D is a value in the range: $[0, 1]$ representing the average minimum distance of vehicles to obstacles, and C and B are integers representing the number of collisions and failed vehicle spawning (due to vehicles not moving from a start point), respectively. As the average speed (S) is multiplied by the largest weight, vehicles get the most fitness by constantly moving and getting to an end-point on the road network. The distance (D) variable encourages vehicles to keep a safe distance between them, making collisions less likely.

C. Task Performance Evaluation

Methods for automated intersection management (sections III-A and III-B) were evaluated and compared as follows. The AIM controller was run on each of the 10 road networks and an average task performance (over 20 runs) calculated for each of the task performance metrics (table II).

For NEAT, the fittest controller was selected from 20 evolutionary runs and set as the vehicles' controller on each of the 10 tracks, where the same evaluation procedure as AIM was used. That is, the evaluation of the fittest NEAT evolved controller (for a given road network) was done in 20 non-adaptive simulation runs, where average task performance was calculated for each of the task performance metrics (table II).

IV. Results & Discussion

Figures 2 and 3 present results, averaged over 20 runs, for each method evaluated on all road environments for each task performance metric (table II). Specifically, figure 2 presents average vehicle throughput, where as figure 3 presents average vehicle speed across all road networks tested. In all figures, the metric has been normalized where values of 0.0 and 1.0 indicate the minimum and maximum values, respectively, for the given metric in a simulation task trial. Overall results indicate that AIM using the FCFS protocol for intersection traffic management yields a high average task performance for average vehicle throughput (table II).

However, using the NEAT method for automating intersection management out-performs AIM on specific types of road networks and otherwise yields comparable task performance. For example, given average vehicle throughput (figure 2), a pair-wise comparison between the fittest NEAT evolved controller and AIM, using the *Mann Whitney U* test ($p < 0.05$) [30], indicated that the NEAT evolved controllers yielded a significantly higher average vehicle throughput on road networks 3, 6 (figure 4) and 10 (figure 5). Where as, for all other road networks NEAT and AIM controllers yielded a comparable average vehicle throughput.

That is, the AIM controller attained a 100% throughput rate on seven of the road networks, but did not produce

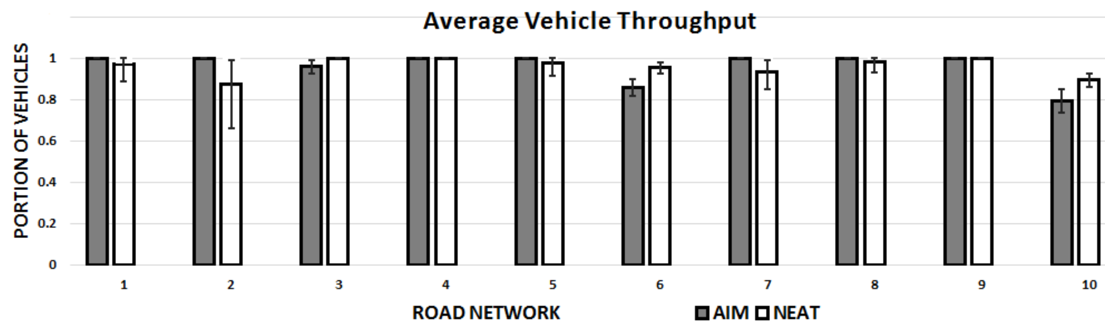

Fig. 2. Vehicle throughput evaluation on each road network: Average portion of vehicles (over 20 runs) that arrived at their destination.

optimal throughput for road networks 3 (traffic-circle), 6 (eight-way intersection) and 10 (double lane merge and one-way intersection exits). Whilst previous work [7] demonstrated that AIM directs vehicle traffic optimally in three or four-way intersections with single or dual carriage lanes, such as those of the other road networks (figures 4 and 5), intersections that use traffic circles, many more entry and exit points including one way roads and multi-lane roads that merge have received relatively little attention in AIM focused research.

Average vehicle throughput results (figure 2) indicate that the AIM controller (using the FCFS protocol) is not as well suited to handling networks of intersections that include such types of road features. In support of this consider the average speed (figure 3) for the AIM controller for all road-networks. A pair-wise statistical comparison between the fittest NEAT evolved controller and AIM controller for all road networks, using the *Mann Whitney U* test ($p < 0.05$), indicated that, with the exception of road network 9, NEAT evolved controllers yielded a significantly higher average vehicle speed.

The relatively lower average speed of the AIM controller on all road networks, except road network 9, and the higher average vehicle throughput of NEAT evolved controllers on road networks 3, 6 and 10, is indicative of the AIM controller not directing vehicles to enter an intersection until a path with no collisions can be guaranteed. All road networks are conducive to high traffic flow meaning that vehicles entering an intersection will on average wait for longer periods before they have a clear path. For example, there is a one-way path around the traffic circle (road network 3, figure 4), thus vehicles waiting to enter a traffic circle must wait for a sufficiently large gap in the traffic in order that a clear path can be followed. Also, the intersections in road network 10 (figure 5) have seven points of entry from one way roads, meaning that on average delays and reduced vehicle speeds will result if there is high volume of vehicles as vehicles will tend to be lined up on the one-way road entry points, waiting for a clear path. The structure of the four-way intersections of road network 6 (figure 4) yielded similar problems for the AIM controller resulting in reduced average vehicle speeds.

In terms of average vehicle speed, road network 9 (figure 5) was the exception, where there was no significant difference between the average speed of the fittest NEAT evolved controller and the AIM controller. The exact features of road network 9 that resulted in comparable average speeds for both methods remains a topic in ongoing research.

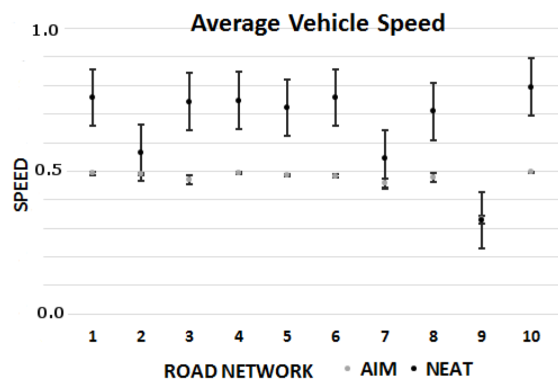

Fig. 3. Average speed (20 runs, all vehicles) per method per road network.

In terms of vehicle throughput, the fittest NEAT controller, evolved on each of the road networks (figures 4 and 5), was able to coordinate vehicles with a comparable task performance, compared to the AIM heuristic controller, with the exception of road networks 3, 6 and 10. The significantly higher average vehicle throughput of the fittest NEAT evolved controllers on road networks 3, 6 and 10 and the higher average vehicle speed on all road networks, except road network 9, is theorized to result from the decentralized coordination of NEAT evolved behaviors. That is, NEAT controllers evolved sensory-motor correlations such that all vehicles moved collectively and in close proximity to each other when passing through intersections.

The emergence of this collective behavior was directed by the fitness function (section III-B1) that selected for driving behaviors that maximized vehicle throughput (movement between road network entry and exit points) and thus vehicle speed. Counter to this, the structure of the road networks tested (figures 4 and 5) forced AIM controlled vehicles to queue at intersections given the high volume of traffic that was tasked to pass through the intersections of these road networks, thus lowering average vehicle speed and throughput.

Thus, in the case of intersections with many entry and exit points and connecting one-way roads (such as road networks 3, 6, and 10), the NEAT evolved controller was better able to handle increased traffic flow and traffic congestion in the intersections. Figure 1 (right) presents the ANN topology of the fittest NEAT controller evolved in road network 10. This example fittest controller supports previous work [31], [32], indicating that NEAT is a suitable method for deriving

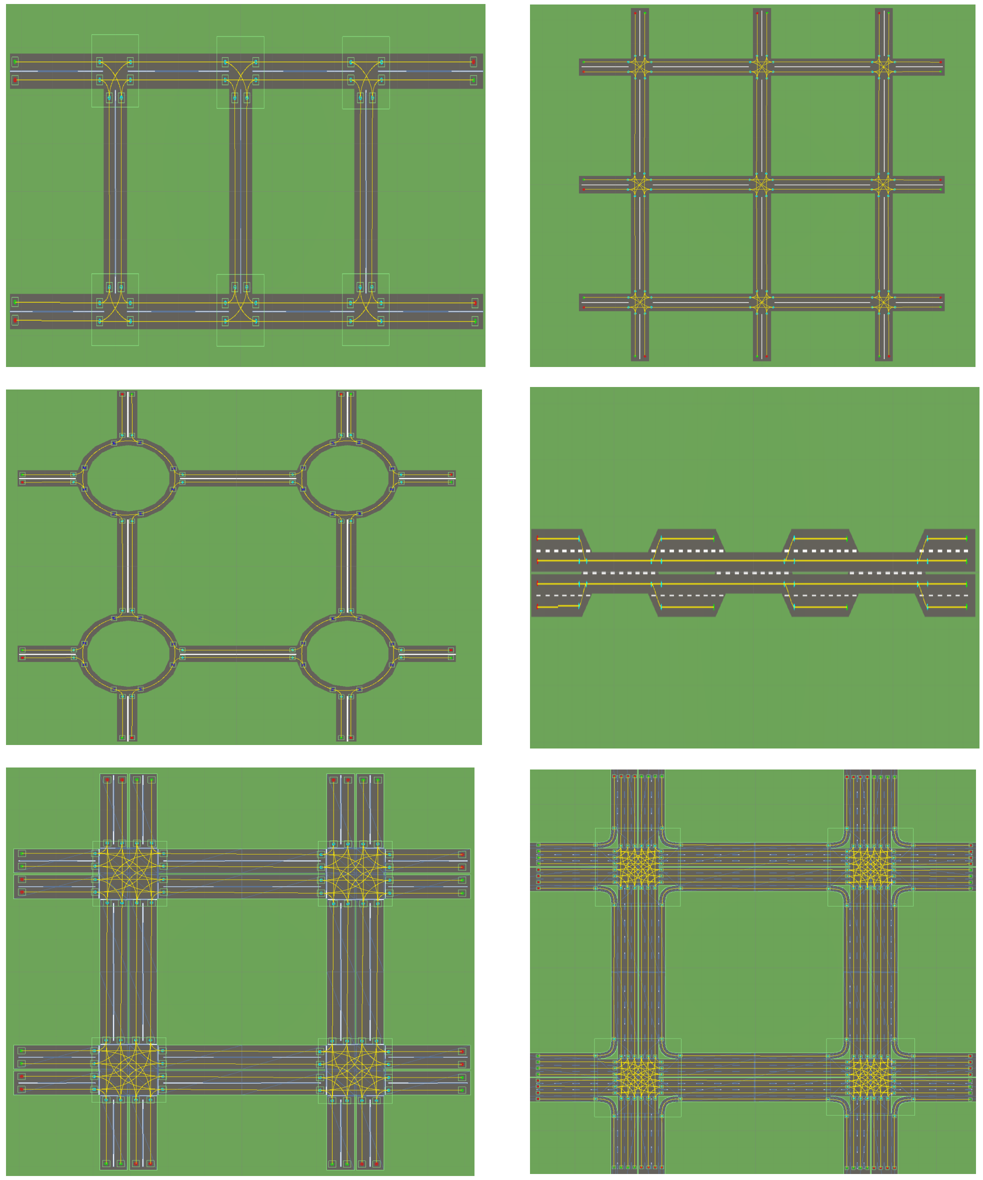

Fig. 4. Road networks 1 to 6 on which the fittest NEAT evolved versus an AIM heuristic controller was tested (road network 1 is top left, road network 6 is bottom right). On road networks 3 and 6 AIM controllers yielded a lower speed and throughput, compared to the fittest NEAT evolved controllers.

controllers that encapsulate enough functional complexity to generalize to task variants. In this study, the fittest NEAT controllers were complex enough such that sensory inputs could be appropriately mapped to motor outputs for various vehicle configurations (positions, orientation and speed) in the intersections on any given road network (figures 4 and 5).

It is important to note that very occasionally vehicle collisions did occur during the evaluation of the fittest NEAT evolved controllers on some road networks. Specifically, there was on average less than one collision on road networks 1 and 6 (figure 4), and road networks 7, 8 and 10 (figure 5). In this case the AIM controller had a clear advantage of yielding no collisions for all road networks tested. The issue of reducing collisions to zero for any given road network must clearly be addressed if NE is to be implemented for vehicle control in automated intersection management. One potential contingency is to combine evolved ANN controllers with heuristics specifically designed for collision avoidance that activate in imminent hazardous situations where the ANN is predicted to fail. That is, when the ANN has failed in simulated tests on similar road networks or when a safety threshold is exceeded. For example, too high a speed given an obstacle in the vehicle's path or too close of a distance between vehicles and obstacles. However, such hybrid methods and their evaluation remains the topic of ongoing research.

In these simulations, NEAT was able to leverage few of the benefits associated with using NE to adapt vehicle controllers. That is, NE is best suited to evolve controllers to adapt to dynamic, noisy task environments, where controllers must process incomplete sensory information [13], [33], into appropriate motor outputs. Importantly, such conditions were not present in the task environments (road networks) tested in this study. That is, the intersection management task assumed that there was no vehicle sensor noise or sensor failures, no

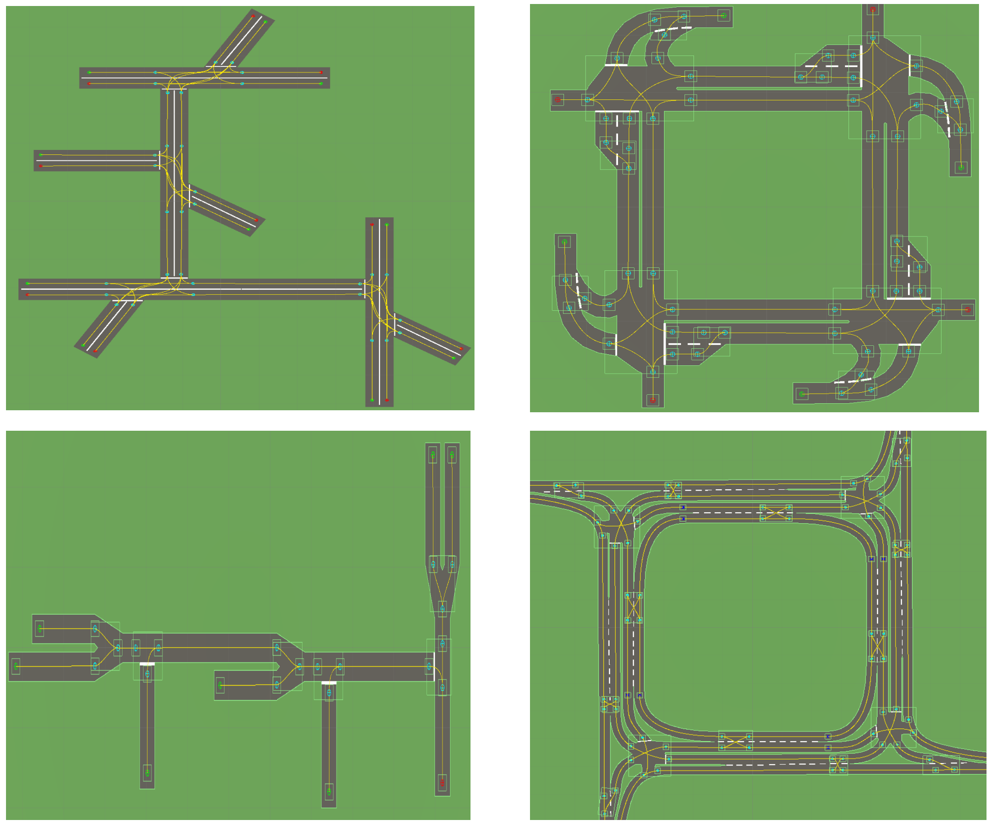

Fig. 5. Road networks 7 to 10 on which the fittest NEAT evolved versus an AIM heuristic controller was tested (road network 7 is top left, road network 10 is bottom right). On road network 10 AIM controllers yielded a lower speed and throughput, compared to the fittest NEAT evolved controllers.

uncertainty in vehicle operations (such as mechanical failures [8]), and no unpredictability in traffic conditions (such as pedestrians). Intersection management tasks with these types of conditions favor an AIM controller, which makes such assumptions about the task in order for AIM controlled vehicles (using the FCFS protocol) to operate optimally.

Thus, this study's results corroborate the benefits of using an AIM controller with the FCFS protocol for specific types of intersections [2], [6], [7], though also demonstrate the efficacy of using NE to adapt vehicle controllers for automating intersection management. That is, NEAT evolved controllers yielded significantly higher task performance in terms of average vehicle speed for nine of the ten tested road networks, significantly higher vehicle throughput on three road networks, and comparable vehicle throughput on the other road networks.

To the best of the authors' knowledge this is the first study that has compared AIM (with the FCFS protocol) as a centralized heuristic based approach, with NE evolved controllers, as a decentralized evolutionary approach. An important caveat to this study was that it assumed the vehicles operated in perfect traffic conditions. That is, there was no sensory noise or unpredictability such as obstacles on the roads or pedestrians crossing roads or intersections in contravention of traffic laws.

Thus, in order to emulate the constant uncertainty of real road traffic conditions, current work on this topic is investigating the efficacy of NE for evolving controllers given increasing levels of unpredictable behavior on road networks. For example, uncertainty will be introduced in the form of sensor noise, pedestrians crossing roads and intersections at random locations as well as obstacles appearing on the roads.

V. Conclusions

This study presented a comparative evaluation of two distinct approaches to automated intersection management for groups of autonomous vehicles in road networks of connected intersections. The first was a centralized heuristic based controller that used the AIM system with the FCFS protocol. The second was a decentralized NE approach that adapted vehicle controllers as they collectively navigated an intersection. The study's objective was to ascertain the efficacy of both approaches for a broad range of road networks, where there were no constraints of traffic lights or stop signals at intersections. Results indicated that NE evolved controllers were an effective method for automating coordinated intersection management behaviors in groups of autonomous vehicles for a broad range of road networks. Also, even though the task assumed perfect traffic flow conditions and vehicle sensory information, NE evolved controllers out-performed AIM controlled vehicles (in terms of maximizing vehicle speed and throughput, thus minimizing delays) on intersections comprising many roads or intersections connected to one-way roads.

References

[1] M. Middlesworth, K. Dresner, and P. Stone, "Replacing the stop sign: Unmanaged intersection control for autonomous vehicles," in *Proceedings of the The Fifth Workshop on Agents in Traffic and Transportation Multiagent Systems*, Estoril, Portugal, 2008, pp. 94–101.

[2] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management," *Journal of Artificial Intelligence Research*, vol. 31, no. 1, pp. 591–656, 2008.

[3] D. Fajardo, A. Tsz-Chiu, T. Waller, P. Stone, and D. Yang, "Automated intersection control: Performance of a future innovation versus current traffic signal control," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2259(1), pp. 223–232, 2012.

[4] C. Fok, M. Hanna, S. Gee, T. Au, P. Stone, C. Julien, and S. Vishwanath, "A platform for evaluating autonomous intersection management policies," in *Proceedings of the ACM/IEEE Third International Conference on Cyber-Physical Systems*. Beijing, China: IEEE Press, 2012, pp. 87–96.

[5] M. Quinlan, T. Au, J. Zhu, N. Stiurca, and P. Stone, "Bringing simulation to life: A mixed reality autonomous intersection," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. San Francisco, USA: IEEE Press, 2011, pp. 6083–6088.

[6] M. Hausknecht, T. Au, and P. Stone, "Autonomous intersection management: Multi-intersection optimization," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. San Francisco, USA: IEEE Press, 2011, pp. 4581–4586.

[7] T. Au, N. Shahidi, and P. Stone, "Enforcing liveness in autonomous traffic management," in *Proceedings of the Twenty-Fifth Conference on Artificial Intelligence*. San Francisco, USA: AAAI Press, 2011, pp. 1317–1322.

[8] T. Au, C. Fok, S. Vishwanath, C. Julien, and P. Stone, "Evasion planning for autonomous vehicles at intersections," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vilamoura, Portugal: IEEE Press, 2012, pp. 1541–1546.

[9] T. Au, M. Quinlan, and P. Stone, "Setpoint scheduling for autonomous vehicle controllers," in *Proceedings of the IEEE International Conference on Robotics and Automation*. St. Paul, USA: IEEE Press, 2012, pp. 2055–2060.

[10] K. Dresner and P. Stone, "Sharing the road: Autonomous vehicles meet human drivers," in *Twentieth International Joint Conference on Artificial Intelligence*, 2010, pp. 1263–1268.

[11] T. Au, S. Zhang, and P. Stone, "Autonomous intersection management for semi-autonomous vehicles," in *Handbook of Transportation*. Routledge, USA: Taylor and Francis Group, 2015, pp. 1451–1452.

[12] P. Ross, "Europe's smart highway will shepherd cars from rotterdam to vienna," *IEEE Spectrum*, vol. December, 2014.

[13] D. Floreano, P. Dürr, and C. Mattiussi, "Neuroevolution: from architectures to learning," *Evolutionary Intelligence*, vol. 1, no. 1, pp. 47–62, 2008.

[14] S. Baluja, "Evolution of an artificial neural network based autonomous land vehicle controller," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 3, pp. 450–463, 1996.

[15] J. Drchal and J. Koutník, "HyperNEAT controlled robots learn how to drive on roads in simulated environment," in *Proceedings on the IEEE Congress on Evolutionary Computation*. Trondheim, Norway: IEEE Press, 2009, pp. 1087–1092.

[16] K. Stanley, N. Kohl, R. Sherony, and R. Miikkulainen, "Neuroevolution of an automobile crash warning system," in *Proceedings of the Conference on Genetic and Evolutionary Computation*. Washington DC, USA: ACM Press, 2005, pp. 1977–1984.

[17] W. van Willigen, E. Haasdijk, and L. Kester, "A multi-objective approach to evolving platooning strategies in intelligent transportation systems," in *Proceedings of the Genetic and Evolutinary Computation Conference*. Amsterdam, the Netherlands: ACM Press, 2013, pp. 1397–1404.

[18] F. Ruini and A. Cangelosi, "Extending the evolutionary robotics approach to flying machines: An application to MAV teams," *Neural Networks*, vol. 22(1), pp. 812–821, 2009.

[19] M. Salichon and K. Tumer, "A neuro-evolutionary approach to micro aerial vehicle control," in *Proceedings of the Genetic and Evolutinary Computation Conference*. Portland, USA: ACM Press, 2010, pp. 1123–1130.

[20] S. Hauert, J. Zufferey, and D. Floreano, "Evolved swarming without positioning information: An application in aerial communication relay." *Autonomous Robots*, vol. 26, no. 2, pp. 21–32, 2009.

[21] T. Praczyk, "Using augmenting modular neural networks to evolve neuro-controllers for a team of underwater vehicles," *Soft Computing*, vol. 18, no. 12, pp. 2445–2460, 2014.

[22] M. Duarte, V. Costa, J. Gomes, T. Rodrigues, F. Silva, and S. Oliveira, "Evolution of collective behaviors for a real swarm of aquatic surface robots," *PLoS ONE*, vol. 11(3), p. e0151834.doi:10.1371/journal.pone.0151834, 2016.

[23] K. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary Computation.*, vol. 10, no. 2, pp. 99–127, 2002.

[24] D. D'Ambrosio and K. Stanley, "Scalable multiagent learning through indirect encoding of policy geometry," *Evolutionary Intelligence Journal*, vol. 6, no. 1, pp. 1–26, 2013.

[25] J. Togelius and S. Lucas, "Evolving controllers for simulated car racing," in *Proceedings of the IEEE Congress on Evolutionary Computation*. Edinburgh, UK: IEEE Press, 2005, pp. 1906–1913.

[26] ——, "Evolving robust and specialized car racing skills," in *Proceedings of the IEEE Congress on Evolutionary Computation*. Vancouver, Canada: IEEE Press, 2006, pp. 1187–1194.

[27] J. Togelius, P. Burrow, and S. Lucas, "Multipopulation competitive co-evolution of car racing controllers," in *Proceedings of the IEEE Congress on Evolutionary Computation*. Singapore: IEEE Press, 2007, pp. 4043–4050.

[28] L. Cardamone, D. Loiacono, and P. Lanzi, "Learning to drive in the open racing car simulator using online neuroevolution," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2(3), pp. 176–190, 2010.

[29] L. Cardamone, A. Caiazzo, D. Loiacono, and P. Lanzi, "Transfer of driving behaviors across different racing games," in *Proceedings of the IEEE Conference on Computational Intelligence and Games*. Seoul, South Korea: IEEE Press, 2011, pp. 227–234.

[30] B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes*. Cambridge, UK: Cambridge University Press, 1986.

[31] J. Moore and A. Clark, "Hold the spot: Evolution of generalized station keeping for an aquatic robot," in *Proceedings of the Fourteenth International Conference on the Synthesis and Simulation of Living Systems*. East Lansing, USA: MIT Press, 2014, pp. 200–201.

[32] R. Koppejan and S. Whiteson, "Neuroevolutionary reinforcement learning for generalized control of simulated helicopters," *Evolutionary Intelligence*, vol. 4, no. 1, pp. 219–241, 2011.

[33] K. Stanley, D'Ambrosio, and J. Gauci, "Hypercube-based indirect encoding for evolving large-scale neural networks," *Artificial Life*, vol. 15, no. 1, pp. 185–212, 2009.