Using Multiple Worlds for Multiple Agent Roles in Games

Joseph Alexander Brown
Artificial Intelligence in Games Development Lab
Innopolis University Innopolis
Republic of Tatarstan, Russia
j.brown@innopolis.ru

Daniel Ashlock
Department of Mathematics and Statistics
University of Guelph
Guelph, Ontario, Canada, N1G 2W1
dashlock@uoguelph.ca

***Abstract*—In a game it is often the case that there are multiple roles or types of actors with different goals. One possible target for automatic content generation is to create multiple different software agents for these distinct roles. This paper outlines a technique, based on the multiple worlds model, for creating such actors via evolution. The objective function is based on the performance of the actors within their role and retains the ability of evolution to operate on populations of agents, permitting the creation of many possible agents for each role. The fitness of distinct agent types is never compared by this algorithm, so the evolution of agents affects fitness only at the level of the user specified simulation used to evaluate performance. The technique is demonstrated by simultaneously evolving four populations of prisoner's dilemma agents with very different roles.**

I. Introduction

The origin of the proposal made in this paper was a failed attempt at modeling an electric power market. The system being modeled included companies that generate power, companies that transmit the power to different industries, and companies that sell the power locally in municipalities or rural areas. The point at which this simulation failed was that a single encoding was used for all three types of actors and these actors were evaluated in (and had to perform well in) all three different roles. The decision to proceed in this fashion was based on a desire for simplicity, but this turned out to be a case of false economy.

In this paper a vision is outlined for training agents with distinct, even very different roles within a game. This technique is called the *multiple worlds model* (MWM) [13]. A simple motivating example is used: suppose that you are entering a prisoner's dilemma tournament and want several collaborators to submit strategies that will roll over and play dead to help your entry. Suppose, also, that this is against the contest's rules, at least if it is detected. An approach to creating stealth collaborators might work as follows. A training system is used to evolve prisoner's dilemma agents with the distinct roles. The first and second types of agents simply try to maximize their own score. The third attempts to maximize the score of the first type and minimize the score of the second, ignoring its own score. Once training is finished the third agent type is what is submitted as a hidden collaborator. The fact that this collaborator is making an honest effort to beat agents of the second type, and because of that training some additional agent types more or less accidentally, means that the secret collaborator will not simply role over an play dead. If greater stealth several agents with the role of the second agent could be included encouraging a greater variety of stealth behavior in the collaborator agent.

The authors thank the University of Guelph and Canadian Natural Science and Engineering Research Council of Canada (NSERC) for supporting this work.

An experiment in this paper shows that a situation like the motivating example can be created using the proposed MWM system. The application of simultaneous training of agents with different roles goes far beyond this example. Creating distinct types of non-player characters in an adventure game, load outs with magical interactions in role playing games, interesting opponents in a first person shooter, or even different types of puzzles or mini-games could all be accomplished with this type of training system.

The remainder of this study is organized as follows. Section II covers the background on software agents that interact with one another and their training. Section III explains the multiple worlds approach and looks at possible alternative methods. Section IV outlines the small demonstration experiment. Section V demonstrates that setting the agents different goals caused them to develop very different strategies. Section VI outlines different ways the multiple worlds algorithm could be applied to the problem of creating populations of agents with well defined roles.

II. Background

Training multiple types of agents for a game is not a new idea. An early approach to interesting collective behavior appeared with the *boids* [35]. This simulation treated the flocking behavior of birds or herd motion in animals. Each agent used the same set of rules, but those rules were based on the density and behavior of nearby agents. This approach is a good one, but only when the agents share a single type of behavioral role.

Another approach is to create a multi-agent system in which the agents have cooperative roles in solving a problem [34]. The algorithm proposed here is another approach to this technique. Assigning different parts of a task to different agents is a method of outsourcing the task of decomposing a problem to evolution – an excellent thing when it works. The MWM model is capable of supporting this sort of cooperative multi-agent problem decomposition.

Recent approaches within games [32] propose to use multi-criteria optimization to evaluate the agent's behavior. This is problematic simply because multi-criteria optimization is, itself, quite hard. The system outline in this paper does not use multi-criteria optimization; rather it uses multiple criteria optimization. Each type of agent has its own fitness based on the goals associated with its role; since these fitness criteria are never compared to one another, the problematic issues of multi-criteria optimization do not arise. The system proposed here, while leaving the behavior evaluations intertwined, acts to decompose the problem of agent training.

Another type of training system with multiple roles, albeit shared roles, are those that learn games by self play. Systems for backgammon [43], checkers [18], chess [17], and Othello (also called Reversi) [30] have all enjoyed success. There were reasons to think that self-learning systems would fail because of a tendency to develop a "local culture" that causes the population of agents to wander off into a part of the strategy of no interest to serious players. It turns out that this problem can be avoided at the design level of the training system and game playing agents like the *Blondie* chess and checkers players, trained in this fashion, are commercially available.

A famous example of multiple-role evolution is the evolution of sorting networks in [26] in which the trial sequences also evolved to exploit any inadequacies in the evolving sorting networks. In this case one set of evolving actors had the job of testing the other. The technique was found to enhance the performance of evolution and subsequently the question of tuning the parasite reward was explored [15]. Applications of the idea of co-evolving test cases was applied to a different problem in [5]. These techniques have not yet been applied to procedural content generation, but there is clearly room for such an application.

Training game playing agents by self-play is a natural research domain to extend with the MWM. It provides an additional avenue from preventing the agents from wandering into an odd section of the strategy space. If such a strategy-wander occurred then it would represent an exploitable weakness in the agents. The use of multiple, genetically isolated populations, as in the MWM, would cause the system to optimize against and prevent such weaknesses from emerging. The genetic isolation is critical as it prevents genetic collaboration that would move the population to odd strategy pockets.

The notion of training non-player characters with a multi-agent system has been tested [27] using a neural net approach. Other natural application for multi-agent systems include scheduling and routing [38], cooperative versions of the pursuit problem [14] in real-world and game situations, and even modeling the emergence of civil violence with game theory [20].

All of these multi-agent training systems are potential targets for research with the MWM system. The ability to write role-specific objective function when using the MWM system means that there is good hope of being able to improve performance, e.g. to have instigator, enforcement, and crowd agents with distinct goals and characters arising from evolution when trying to model the emergence of civil violence.

A. Evolutionary Game Theory

The motivating example for the MWM presented in this study is based on the evolution of agents for a mathematical game. This places it within the domain of evolutionary game theory, which we now review and critique. In many game theoretic situations, initially a player (or an agent) tries to work out what is best for itself considering all possible scenarios. However, in new situations, often it is uncertain how others will play the game and so the decisions made by the player are myopic since players cannot fully compute optimal strategies for others. Players, therefore, adapt and respond to situations through learning. When that method of learning is evolution, we arrive at the field of evolutionary game theory.

In the simplest settings, each organism plays a particular strategy where, then, it is rewarded with payoffs or fitness. Rarely, some offspring randomly play one of strategies when mutations occur. Evolution is typically modeled using two different approaches, through competition, proposed by [41], and by replicator dynamics [45]. Replicator dynamics models competition via differential reproduction. Both techniques for modeling the evolution of strategies over time attempt to predict the outcome of evolution by locating Nash equilibria [33] or evolutionary stable strategies [40]

1) Predication Fails: An evolutionary stable strategy (ESS) is one that, if adopted by a population, cannot be invaded by a single instance of any other strategy. Evolutionary stable strategies are taken to be attracting states of an evolutionary training system but seldom are. There are several reasons for this.The first is that definition treats only invasion by a single agent but most evolutionary training systems update multiple agents simultaneously. Failure to meet the hypothesis of a definition may not be problematic, but it at least merits additional scrutiny. Another serious problem, documented in [19] addresses the issue that the theoretical structure supporting ESS requires an infinite population which is not realistic. Experiments performed by Fogel demonstrate that finite populations do not discover the ESS for the hawk-dove game.

Nash equilibria also have problems. The definition of a Nash equilibria is a collection of strategies with the property that unilateral change of strategy by any player decreases their score. This means that Nash equilibria also suffer from the assumption that only one player changes at a time, a false-to-fact assumption in most evolutionary agent training systems. Nash equilibria also neglect the problem of neutral networks. Changes in agents that do not affect payoffs when they happen are allowed; this permits agents to drift to an exploitable state, e.g. mutation from tit-for-tat to always cooperate, so that a set of strategies that appear to be in a state of Nash equilibria can use genetic drift to escape.

We raise the issue of these tools for predicting the stable states of evolutionary processes in part to examine them in the context of training agents to multiple roles. In such a training environment, unless we take the extremely restrictive step of updating only one agent within each role at a time before re-evaluating fitness, we violate the hypothesis needed for both ESS and Nash equilibria. In essence, use of the MWM, which offers a number of benefits makes the problem of inapplicability of both Nash equilibria and ESS worse. It seems likely that other, more general, notions of stability will be needed in any theoretical analysis of a multiple-role agent training system.

B. The Prisoner's Dilemma

The demonstration of the multiple agent training protocol in this paper is based on a much studied game. The prisoner's dilemma [8], [7] is a classic model in game theory. Two agents each decide, without communication, whether to cooperate (C) or defect (D). The agents receive individual payoffs depending on the actions taken. The payoffs used in this study are shown in Figure 1. The payoff for mutual cooperation C is the *cooperation* payoff. The payoff for mutual defection D is the *defection* payoff. The two asymmetric action payoffs S and T, are the *sucker* and *temptation* payoffs, respectively. In order for a two-player simultaneous game to be considered prisoner's dilemma, it must obey the following pair of inequalities:

$$S \leq D \leq C \leq T \tag{1}$$

and

$$2C \leq (S + T). \tag{2}$$

In the *iterated prisoner's dilemma* (IPD) the agents play many rounds of the prisoner's dilemma. IPD is widely used to model emergent cooperative behaviors in populations of selfishly acting agents and is often used to model systems in biology [39], sociology [28], psychology [36], and economics [25].

(1)

		$\mathcal{S}$	
		C	D
$\mathcal{P}$	C	3	5
	D	0	1

(2)

		$\mathcal{S}$	
		C	D
$\mathcal{P}$	C	C	T
	D	S	D

Fig. 1. (1)The payoff matrix for prisoner's dilemma used in the demonstration of multiple agent role evolution in this study – scores are earned by strategy $\mathcal{S}$ based on its actions and those of its opponent $\mathcal{P}$ (2) A payoff matrix of the general two player game – C, T, S, and D are the scores awarded.

In this paper the IPD will be used as a simple game to demonstrate training of distinct types of agents via co-evolution. The agents will be differentiated by assigning them distinct tasks, one agent type will be maximizing its own score while another will act as a collaborator maximizing the score of another agent type.

III. Multiple Worlds

The MWM [10], much like island models [46], uses multiple populations of evolving structures. In this case a generational genetic algorithm, see Figure 2 for pseudocode. Unlike island models these populations do not exchange

```
for some number of generations do
  for N Reassortments do
    Randomize the worlds
    for all worlds do
      Run PD game
    end for
  end for
  for all populations do
    Select breeding pairs based on fitness
    Apply Crossover/Mutation
  end for
end for
```

Fig. 2. Pseudocode of the Multiple Worlds system

members, making them genetically isolated. The populations obey the conditions of the biological species concept. The organisms are genetically isolated and able to breed within kind, making them distinct species. Each of these populations/species are evaluated in a competitive fitness function, in which one member from each population is selected, this is a hypothetical *world*, and the group is evaluated on the same set of problem data examples. The population member which is most representative, e.g. least error, of the problem data example, scores fitness for that example.

Such a process is a biological anthology to adaptive radiation effects seen in number of biological organisms. The canonical example being *Darwin's Finches*, or Geospizinae, in the Galápagos Archipelago, discovered by Darwin on the voyage of the H.M.S. *Beagle* (1831–1836). While ignored by Darwin [16], partially due to missclassifcation [42], the works of Lack [31] posit that the beaks of the finch species transmute dependent upon food sources and competition between species types. On islands where more than one species exists, beaks specialize to exploit a single food source. Conversely, on islands with one species the beak generalizes in order to exploit all types of food. Lack's hypothesis was later confirmed in a number of larger long ranging studies (e.g. [21], [22], [23]), which tracked the finches and the seeds on the various islands for a number of years, these studies demonstrated that such niching between species could develop within the course of a single generation based on available food sources.

Furthermore, behavioral modification can also lead to specialization in food sources through a process of niche partitioning. Hanson in *Feathers* gives an anecdotal account of studying the behaviours of North American bird actions in a forest: "Nuthatches foraged mostly on the trunks, Chickadees dominated the main branches, and Kinglets spend their time flitting about in the side branches"[24]. The MWM aims to use such principles of inter-population competition with intra-population evolution to guide a process of partitioning data or agents into model families.

The niche effect is intrinsic to MWM; it does not require an explicit calculation of phenotype distance or a crowding measure. The novel mixed populations method of fitness evaluation implicitly makes crowding undesirable. It has also been shown to increase the diversity of solutions located [13]. This diversity has been seen in studies of mixture vs. monoculture plants in [47], which examined the results of eight years of experimental growth in Jena, Germany. It showed there was an increased interspecific difference to those plants grown in mixture types compared ($p - value < 0.05$) and intraspecific distance within mixture types on traits was increased ($p - value = 0.101$). They attribute a difference in relative specific leaf area ($p - value = 0.073$) and height ($p - value = 0.074$) to specialization into a niche. While these findings where marginally significant correlations, the authors claim that these traits are representative of relevant niche dimensions, and that further study is warranted looking at the processes of change. Tilman and Snell-Rood [44] examine this study, and the previous mentioned finch studies, to question if such studies can experimentally demonstrate a divergence of species.

A. *Previous Applications of Multiple Worlds*

The Multiple Worlds Model (MWM) of Evolution is used in situations where a number of distinct models are required to classify and provide solution to a problem. The first such model examined partitioning of a data space by a number of regressive models[10]. This application is to a technique called *partitioning regression* in which the MWM is used to simultaneously partition a data space and find independent models of each partition. Each members of each population participating in the MWM is rewarded for the data points it models with the lowest error; the populations compete to capture data points. A more general model (one covering more points) is likely to have higher error and so use of the MWM can provide tight models of sensibly partitioned data. This application of the MWM shares with k-means clustering [29] the problem that the number of models must be chosen in advance, but work in [13] shows how to implement an *extinction* protocol that also permits the number of models to be selected.

An application of the MWM to theoretical biology [4] applied the technique to the simulation of bacterial communities. This study was attempting to model the phenomenon that most bacteria cannot be cultured – their dependence on a multi-type community does not permit them to survive on their own. Potentially interdependent bacterial types were evolved to play a metabolism game with each type assigned to a population withing the MWM. It was found that about 2% of the evolved bacterial agents could exhibit viable levels of metabolism in monoculture; 98% had evolved an analog to the naturally occurring community dependence.

The MWM was also used to partition collections of iterated prisoner's dilemma agents [11], [12] based on their behaviours. On one set of agents, this effort failed in an interesting way leading to the discovery of a new agent type: *trifecta* The trifecta agent plays optimally against a mixed set of opponents consisting of always cooperate, always defect, and tit-for-tat, rather than partitioning them. This failure highlights a strength of the MWM, also observed in some of the data partitioning efforts. If a single population captures all the points in a data analysis or manages to out score the other populations against a diverse collection of agents then MWM has located a previously unsuspected general model.

In a more cooperative application of the MWM, Scirea and Brown [37] demonstrated its application to the creation of four part harmonies meeting with a number of well defined musical rules. The developed populations specialized into altos, tenors, when given a baseline. The fitness was based on the ability for the developed singers to fit the rules of harmony. Hence, specialization was required in order to, but again this was not based on any requirements from the individual fitness and each was evaluated in the same method. This could be added, marking each population as having a range on the singer's abilities in terms of their voice as a constraint. These applications show that the method is suitable in both progressing forward in both cooperative and collaborative games, in which ones own personal score might be regulated by some individual changes.

The MWM has been shown to be able to break down into categories via a process of extinction events within the algorithm. This was demonstrated in [9] where multiple radio stations competed over advertisement revenues from a set of listeners with consumer preferences. When the market differentiation was low, the stations would drift about the state, however, when a new competing station was introduced, they would violently move to push to extremes in order to suit only one particular type of listener. In this case, they all had the same fitness of advertisement revenues as a zero sum game. Though one of these stations perhaps could make a deal with a record label and be scored higher when they plug the song as part of a *payola*.

The examples of previous applications of the MWM, when juxtaposed against the examples of multi-agent systems, suggest that the MWM can be applied beneficially to both cooperative and competitive co-evolution. In addition to implementing explicit, isolation-based (and hence low-cost) niching, the MWM can support competitive acceleration of evolution with co-evolving test cases, competitive co-evolution, and problem decomposition via cooperative co-evolution.

IV. Design of Experiments

This study advocates for the use of the MWM as a technique for training agents to fill distinct roles in a game – a new application of the MWM algorithm. A simple demonstration is performed, evolving agents to play the iterated prisoner's dilemma. The difference between this demonstration and experiments performed in earlier studies is that each of four populations of prisoner's dilemma agents is given a distinct goal, realized as fitness functions. In the preponderance of previous experiments training agents to play the IPD, the fitness governing agent evolution was to simply maximize their score [1], [6].

When using the MWM, each agents is tested against the other agents grouped with it. If only one such grouping is used in fitness evaluation, in other words is an agent

TABLE I
GIVEN ARE THE FITNESS FUNCTIONS FOR THE FOUR DIFFERENT AGENT TYPES USED TO DEMONSTRATE TRAINING TO ROLES WITH THE MWM ALGORITHM.

Agent Type	Fitness Function
A	Maximize your own score.
B	Maximize the ratio of scores of agents of type A to agents of Type C.
C	Maximize the ratio of your own score to those of agents of type A.
D	Maximize the average scores of agents of all types.

encountered only one opponent, then the fitness evaluation is potentially non-representative. In this study fitness is evaluated with $k = 6$ re-assortments of the agents before each population updating. This number was selected by preliminary experimentation that sought to balance the increase in representative quality of fitness against the added computational cost of running more prisoner's dilemma tournaments.

Four populations of 100 agents are evolved using the MWM. The populations are assigned different roles by giving them distinct fitness goals, given in Table I. If earlier experiments, where agents simply tried to maximize their own score, the fitness of the evolving population was unstable because fitness depended not on an objective criterion but rather on the mix of agents in the current population. As we will see, the use of multiple roles actually *stabilizes* fitness values.

The agents used in this study are twelve-state finite state machines (FSMs) using the Mealy architecture [3]. They are stored as a linear list of states with the responses and transitions for each state stored together. Crossover is two point, with the states treated as atomic objects. These agents have an initial state and action, used to initialize play; for crossover the initial states and actions are associated with the first state in the machine. Mutation changes one quality of the FSM changing the initial state 5% of the time, the initial action 5% of the time, a transition 40% of the time, and a response 50% of the time. This mutation operator was chosen for agreement with earlier studies to permit comparison. The FSM agents condition their responses on their opponents last action, using it as input.

The evolutionary algorithm used in this study is generational. Selection is performed with single tournament selection with tournament size four. In this model of evolution, the populations of one hundred agents are shuffled into groups of four. The two most fit members of each tournament are copied over the two least fit. The copies are then subject to crossover and mutation. Population updating is continued for 250 generations. Thirty replicates of the MWM algorithm are performed.

A. Analysis Techniques

The hypothesis of this study is that the different agent populations will engage in distinct forms of behavior. In [1] a representation-independent technique called *fingerprinting* was established to create a numerical signature of the behavior of prisoner's dilemma agents. Computing the fingerprint of an agent requires the use of a type of probe strategy called a *Joss-Ann* strategy. The specific probe used to assess the agents in this study is $JA(TFT, \alpha, \beta)$ which generates a cooperate with probability α, and defect with probability β, and otherwise makes the play that a tit-for-tat agent would make. An agent's fingerprint is the expected score that agent would get against $JA(TFT, \alpha, \beta)$, computed using a Markov chain, for 25 pairs of values of α and β.

The fact that fingerprints comprise 25 values makes displaying them challenging. *Evolutionary non-linear projection* [2] is used to map high dimensional data into two dimensions. Evolution is used to maximize the Pearson correlation of the distance matrices of the high dimensional data and its two dimensional surrogate. The projected data has correct *relative* distance but does not preserve scale and so is scaleless.

V. RESULTS AND DISCUSSION

Figure 3 shows the evolution of the population average fitness for four different replicates of the MWM algorithm. Unlike earlier studies where a single population was evolving to maximize its own fitness, this experiment exhibited remarkable similarity between the fitness tracks of different runs. In aggregate, the multi-population system implementing the MWM actually created a situation more stable than the one-population experiments. This suggests that, as an agent training algorithm, the MWM is potentially more reliable for producing agents to fill diverse roles.

The type **D** agents are attempting to maximize the average score of all agents. Since they are playing all other agent types, a clear strategy is for these agents to cooperate into defection from the other agent's defection. The score in all four plots suggests that there are, in fact, doing this. The plots for the **B** and **C** agents is attempting to maximize a ratio of scores; their values stay near one suggesting adaptation by the agent populations to keep the ratios near even. Only the agents of type **A** actually increase their fitness and that only on average.

The key reason for performing this experiment is to establish that agents evolving in different populations learn distinct behaviors. The fitness tracks shown in Figure 3 do *not* show the different fitness values increasing and so a more subtle indicator is needed that the agents are behaving differently. This is where fingerprinting and evolutionary non-linear projection are applied. Figure 4 shows a *non-linear projection* of the fingerprints of the 400 agents from the first run of the MWM algorithm performed.

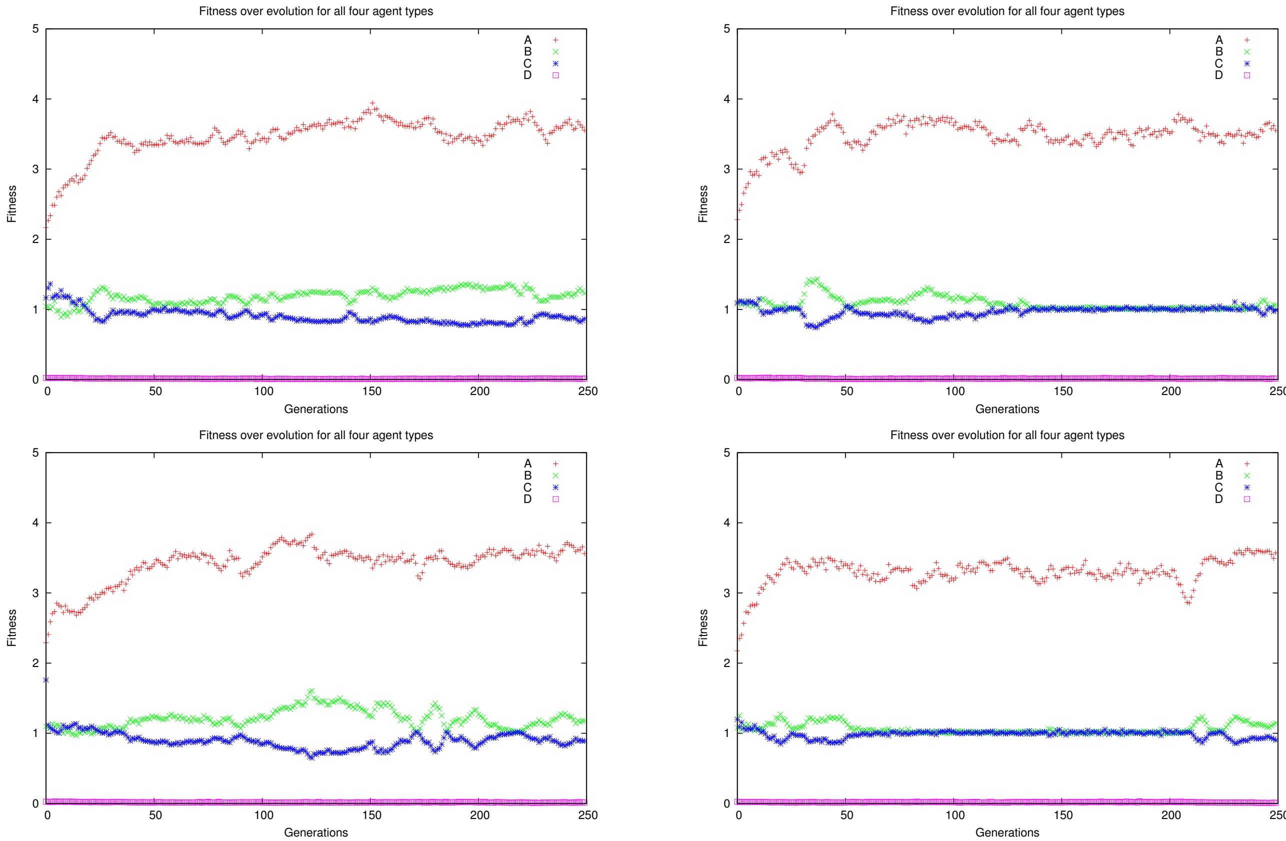

Fig. 3. Shown are exemplary fitness plots giving the average fitness over the course of evolution of the four agent populations.

Figure 4 provides objective information that the agents are evolving to exhibit very different behaviors in the service of their roles. The roll-over-and play agents of type **D** and the type **A** agents that are just maximizing their scores both have a relatively restricted distribution of behaviors. The small number of violet and red outliers represent low-fitness mutants in the evolving population. The agents of type **B** and **C** – the ratio maximizers – exhibit a much wider range of behaviors. Overall Figure 4 demonstrates that the MWM has trained the agents to different roles.

One clear result is that the type **D** agents have learned to increase the other agents scores – if we were using MWM to optimize entrants in a tournament then they exemplify planted collaborators. This experiment uses one of a huge range of possible fitness functions and the IPD is one game of the huge number that are available. This experiment is a proof-of-concept for the vision of simultaneously training multiple types of agents with the MWM.

VI. Conclusions and Next Steps

The paper makes the case for multiple applications of the MWM in games research. A small experiment that checks that the system can specialize agents to different roles for the prisoner's dilemma is presented. The MWM has been used for partitioning regression, game playing agent classification, content generation of music, modeling of ecological situations, and to simulate radio markets. Much of the potential application to games remains to be explored.

The MWM can be considered as an alternate technology for any multiple roles system. Enhancing evolution with evolving test cases, styled as parasites, it not only supported my the MWM but permits greater flexibility in the design of such systems. The tendency of evolving systems to rapidly lose diversity means that a co-evolving population of test cases might only address some of the weaknesses of a population of problem solving agents or address them serially – permitting re-emergence of the problems. This issue might arise in game and non-game applications. The MWM permits this situation to be addressed in a simple problem by having multiple, genetically isolated examples of evolving test cases.

The very natural application of evolving game playing agents with different behaviors that support different game roles is exemplified by the demonstration experiment but has potential far beyond it. Consider a role playing game in which the players are engaged in a hack-and-slash quest for treasure and fighting experience. Hostile NPCs are not hard to program or train – but the behavior of these hostiles with respect to *one another* is a place where additionally nuanced roles could lead to a broader design space for the game. Enemy-of-my-enemy effects can be trained into the agents in the game.

Take for example the synth infiltrations in *Fallout 4* in

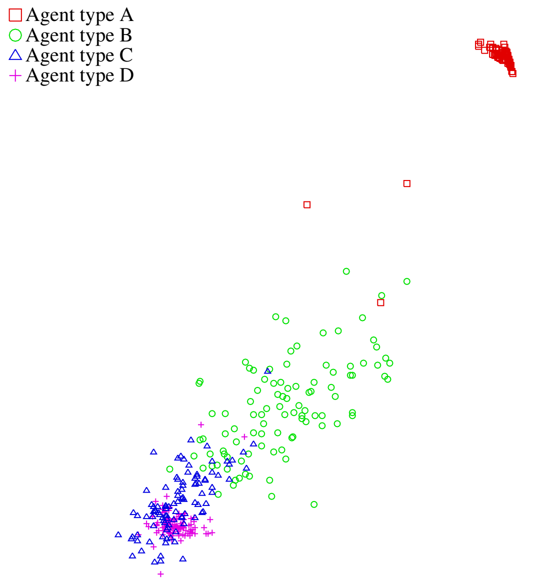

Fig. 4. Shown is a non-linear projection of the fingerprints of 100 PD-playing agents of each of four types from a run of the multiple worlds agent training algorithm. Note that this fingerprinting data indicate that the agent types are substantially different. The Person correlation of the 25-dimensional distances between fingerprints and the intra-point distance of the displayed rendering is $\rho = 0.99388$.

which a game mechanic would allow for synthetic humans to replace settlers in the player villages. If the player is acting in the best interests of the Institute faction, then the synth could stay dormant within the player village and perhaps work to develop the village by taking its utility from the outcome of the player. Conversely, if the player acts against the goals of this faction, the sleeper agent awakens and causes problems in order to satisfy their need for utility. This is currently expressly programmed in *Fallout 4* for the synth to attack the village if the player drops in affinity with the Institute, but could be an emergent outcome of agents trained with the MWM.

If roles can be conditioned on the relative position, success, or power of opponents, then the MWM can be used to train agents that are useful for enhancing game balance. This could be as simple as putting the last place player on a rubber band used in some games; agents with the role of attacking the most dangerous looking or wealthiest looking player. More subtle balance enhancing roles would require careful design of objective functions for those roles.

Another potential application is the development of balance of items in a game. The MWM would allow for the testing of armor combinations if there was some type of affinities in wearing them. For example, there are many RPGs which have bonuses applied to armor sets, having all of the same type of armor. However, there could also be detriments to having mismatched armors, such as having heavy plate greaves and vambrace, while wearing only a leather jerkin about the torso. Or if the world contains magic or daemonic forces, to mix magics of different schools or have both blessed and cursed items near each other in a load out. The magics swirling about the character must be in harmony. Beyond allowing for a character driven environment meeting with narrative, in this application different parts of a suit of armor could be partitioned among the populations of the MWM.

One application of the MWM technique within mathematical games is that of a more complete exploration of the strategy space. Natural collaborator or spoiler agents could be designed by creating fitness function that formed numerical instantiations of those roles; the agents that they collaborate with or attempt to spoil must be present for training, a state of affairs supported by the use of the MWM.

A. *The Cooperation Competition Spectrum*

In the multi-agent systems discussed in the introduction, examples of cooperative and competitive co-evolution both appeared. The partitioning-regression application of MWM is competitive. The music-composition application is cooperative. The demonstration of evolving prisoner's dilemma agents was neither purely cooperative nor purely competitive. Type B agents were cooperating with type A agents in a left-handed fashion while competing with type C agents. Type D agents were trying to cooperate with everyone. The MWM can support competition, cooperation, or any useful mix of these qualities.

References

[1] D. Ashlock, E.Y. Kim, and N. Leahy. Understanding representational sensitivity in the iterated prisoner's dilemma with fingerprints. *IEEE Transactions on Systems, Man, and Cybernetics–Part C: Applications and Reviews*, 36(4):464–475, 2006.

[2] D. Ashlock and A. McEachern. Evolutionary nonlinear projection. *IEEE Transactions on Evolutionary Computation*, 19(6):857–869, 2015.

[3] D. A. Ashlock and E. Y. Kim. Fingerprint analysis of the noisy prisoner's dilemma. In *Proceedings of the 2007 Congress on Evolutionary Computation*, pages 4073–4080, 2007.

[4] Daniel Ashlock and Andrew McEachern. A simulation of bacterial communities. In *2011 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, pages 61–68, 2011.

[5] Daniel Ashlock, Stephen Willson, and Nicole Leahy. Coevolution and tartarus. In *Proceedings of the 2004 Congress on Evolutionary Computation*, volume 2. IEEE Press, Piscataway NJ, 2004.

[6] W. Ashlock. Evolving diverse populations of prisoners dilemma strategies. In *Proceedings of the 2008 Congress on Evolutionary Computation*, pages 1625–1632, Piscataway, NJ, 2008. IEEE Press.

[7] R. Axelrod. *The Evolution of Cooperation*. Basic Books, New York, 1984.

[8] R. Axelrod and W. D. Hamilton. The evolution of cooperation. *Science*, 211:1390–1396, 1981.

[9] J. A. Brown and D. A. Ashlock. Multiple worlds model of evolution for demographic appropriate radio playlists. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8, Dec 2016.

[10] Joseph A. Brown and Daniel Ashlock. Using evolvable regressors to partition data. In *Inteligent Engineering Systems Through Artifical Neural Networks (20)*, pages 187–194. AMSE Press, 2010.

[11] Joseph Alexander Brown. Multiple worlds model for motif discovery. In *2012 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, pages 92–99, 2012.

[12] Joseph Alexander Brown. More multiple worlds model for motif discovery. In *2013 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, pages 168–175, 2013.

[13] Joseph Alexander Brown. *Regression and Classification from Extinction*. PhD thesis, University of Guelph, Guelph, Ontario, Canada, 1 2014.

[14] Y. Uny Cao, Alex S. Fukunaga, and Andrew Kahng. Cooperative mobile robotics: Antecedents and directions. *Auton. Robots*, 4(1):7–27, March 1997.

[15] J. Cartlidge and S. Bullock. Combating coevolutionary disengagement by reducing parasite virulence. *Evolutionary Computation*, 12(2):193–222, 2004.

[16] Charles Darwin. *On the Origin of Species : by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*. Sterling, New York, illustrated edition, 2008.

[17] D. B. Fogel, T. J. Hays, S. L. Hahn, and J. Quon. A self-learning evolutionary chess program. *Proceedings of the IEEE*, 92(12):1947–1954, Dec 2004.

[18] David B. Fogel. *Blondie24: Playing at the Edge of AI*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.

[19] Gary B Fogel and David B Fogel. Simulating natural selection as a culling mechanism on finite populations with the hawk–dove game. *Biosystems*, 104(1):57–62, 2011.

[20] C. K. Goh, H. Y. Quek, K. C. Tan, and H. A. Abbass. Modeling civil violence: An evolutionary multi-agent, game theoretic approach. In *2006 IEEE International Conference on Evolutionary Computation*, pages 1624–1631, 2006.

[21] B. Rosemary Grant and Peter R. Grant. The feeding ecology of darwin's ground finches. *Noticias de Galápogos*, pages 14–18, 1979.

[22] B. Rosemary Grant and Peter R. Grant. Niche shifts and competition in darwin's finches: *Geospiza conirostris and Congeners*. *Evolution*, 36:637–657, 1982.

[23] B. Rosemary Grant and Peter R. Grant. Fission and fusion in a population of darwin's finches: An example of the value of studying individuals in ecology. *Oikos*, 41:530–547, 1983.

[24] Thor Hanson. *Feathers: The Evolution of a Natural Miracle*. Basic Books, 2011.

[25] M. Hemesath. Cooperate or defect? Russian and American students in a prisoner's dilemma. *Comparative Economics Studies*, 176:83–93, 1994.

[26] W. Daniel Hillis. Co-evolving parasites improve simulated evolution as an optimization procedure. *Phys. D*, 42(1-3):228–234, June 1990.

[27] Yaqing Hou, L. Feng, and Y. S. Ong. Creating human-like non-player game characters using a memetic multi-agent system. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 177–184, July 2016.

[28] J. M. Houston, J. Kinnie, B. Lupo, C. Terry, and S. S. Ho. Competitiveness and conflict behavior in simulation of a social dilemma. *Psychological Reports*, 86:1219–1225, 2000.

[29] E.Y. Kim, S.Y. Kim, D. Ashlock, and D. Nam. Multi-k: accurate classification of microarray subtypes using ensemble k-means clustering. *BMC Bioinformatics*, 260(10):1–12, 2009.

[30] K. J. Kim, H. Choi, and S. B. Cho. Hybrid of evolution and reinforcement learning for othello players. In *2007 IEEE Symposium on Computational Intelligence and Games*, pages 203–209, April 2007.

[31] David Lack. *Darwin's Finches*. Cambridge University Press, 1947.

[32] E. Lakshika, M. Barlow, and A. Easton. Understanding the interplay of model complexity and fidelity in multi-agent systems via an evolutionary framework. *IEEE Transactions on Computational Intelligence and AI in Games*, PP(99):1–10, 2016.

[33] J. Nash. Equilibrium points in n-person games. *PNAS*, 36(1):48–49, 1950.

[34] Liviu Panait and Sean Luke. Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems*, 11(3):387–434, November 2005.

[35] Craig W. Reynolds. Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH Comput. Graph.*, 21(4):25–34, August 1987.

[36] D. Roy. Learning and the theory of games. *Journal of Theoretical Biology*, 204:409–414, 2000.

[37] Marco Scirea and Joseph Alexander Brown. Evolving four part harmony using a multiple worlds model. In *Proceedings of the 7th International Joint Conference on Computational Intelligence (IJCCI 2015) - Volume 1: ECTA, Lisbon, Portugal, November 12-14, 2015.*, pages 220–227, 2015.

[38] Franciszek Seredynski. Competitive coevolutionary multi-agent systems: The application to mapping and scheduling problems. *Journal of Parallel and Distributed Computing*, 47(1):39 – 57, 1997.

[39] K. Sigmund and M. A. Nowak. Evolutionary game theory. *Current Biology*, 9(14):R503–505, 1999.

[40] J. M. Smith. The theory of games and the evolution of animal conflicts. *Journal of Theoretical Biology*, 47(1):209221, 1974.

[41] J. Maynard Smith. The theory of games and the evolution of animal conflicts. *Journal of Theoretical Biology*, 47(1):209 – 221, 1974.

[42] Frank J. Sulloway. Darwin and his finches : The evolution of a legend. *Journal of the History of Biology*, 15(1):1–53, Spring 1982.

[43] G. Tesauro. Td-gammon, a self-teaching backgammon program, achieves master-level play. *Neural Computation*, 6(2):215–219, 1994.

[44] David Tilman and Emilie C. Diversity breeds complementary. *Nature*, 515(7525):44–45, 2014.

[45] J. N. Webb. *Replicator Dynamics*, pages 165–185. Springer London, London, 2007.

[46] W. Darrell Whitley, Soraya B. Rana, and Robert B. Heckendorn. Island model genetic algorithms and linearly separable problems. In *Selected Papers from AISB Workshop on Evolutionary Computing*, pages 109–125, London, UK, 1997. Springer-Verlag.

[47] Debra Zuppinger-Dingley, Bernhard Schimd, Jana S. Petermann, Varuna Yadav, Gerlinde B. De. Deyn, and Dan F. B. Flynn. Selection for niche differentiation in plant communities increases biodiversity effects. *Nature*, 515(7525):108–111, 2014.