# Evolving effective behaviors to interact with tag-based populations


Osman Yucel
University of Tulsa
osman-yucel@utulsa.edu

Chad Crawford
University of Tulsa
chad-crawford@utulsa.edu

Sandip Sen
University of Tulsa
sandip-sen@utulsa.edu



## Categories and Subject Descriptors

Computing Methodologies [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

## General Terms

Tag-based mechanisms, stereotypes



## ABSTRACT

Tags and other characteristic features that are consistent among groups of animals or humans can be used to determine appropriate response strategies in societies. This usage of tags can be extended to artificial environments, where agents can significantly reduce cognitive effort by reusing strategies for new interaction partners based on their tags. Strategy selection mechanisms developed based on this idea have successfully evolved stable cooperation in games such as the Prisoner's Dilemma game but relies upon payoff sharing and matching methods that limit the applicability of the tag framework. Our goal is to develop a general classification and behavior selection system based on the tag framework. We propose and evaluate alternative tag matching and adaptation schemes for selecting appropriate behavior against any member of a stable society. The mechanisms allow agents to evolve not only the appropriate strategy-specific agent groups, but more importantly the optimal tag for the environment. We show that these mechanisms will allow for robust selection of optimal strategies within a stable environment and analyze the various environments where this approach is effective.


## 1. INTRODUCTION

Agents in open multiagent environments must be able to quickly adapt to new environments and use past, relevant experience to react to new scenarios and choose effective interaction policies with new partners. As in human societies, agents in artificial societies also come with external features that may, in a large majority of cases, suggest social groups they belong to and hence give at least a coarse-level view of their behavioral characteristics, biases and preferences.

For example, businesses often employ the idea of signalling; that is, to make explicit decisions about the business's appearance (such as the area it is located in, the exterior design of the building, etc) it conveys signals to potential clients about the reliability of the business.

Similarly, consider the scenario of interviewers and interviewees in a hiring situation. In that case, the groups of interviewers and interviewees are two interacting populations. Interviewees are looking for jobs relevant to them: this is done by interacting with interviewers and changing themselves while discriminating the pool of available interviewers to those they would prefer to interact with. Interviewers are more static in their decision making – since they have no incentive to appear more attractive to their audience, their preferences and appearance do not change over time. We would like to simulate this circumstance and similar ones.

External features or tags of an agent can then be utilized to both reduce cognitive effort in the strategy selection and to provide a means to classify agent experience. While these generalizations are not always socially optimal, as in social stereotypes, they do provide a key tool for cognition that allows a pragmatic management of the complexity of life [8].

Prior tag mechanisms have shown that cooperation is possible within the tags framework [10]. These mechanisms however are not particularly suited for open environments. Limitations on agent interactions and payoff distribution may incite cooperation, however they restrict the conditions under which agents can learn within the environment. Another shortcoming is that although these mechanisms allow choosing of which partners in the environment to interact with, it does not enable learning appropriate strategies to interact with each and every member. In this paper, we are interested in developing tag-based mechanisms that allow newcomers to a society to adapt both their tag (external appearance) and their matching strategy (who they want to cooperate/defect against). Therefore we will analyze the interactions of agents in open environment where a group of new agents may migrate to a population of stable agents. The goal of these newcomers, then will be to quickly learn the habitual preferences and behavioral biases of the existing population so that they are able to effectively and gainfully interact with them.

To facilitate speedy learning, we assume two basic features in the environment: (a) the stable population has some latent behavioral biases that are at least indirectly reflected in their external, observable features[1], and (b) members of the newly arriving group can independently interact with the members of the stable population and that their relative success or the reward from this interaction are shared with other newcomers, who can then choose to alter their interaction strategies to align with the more successful of their peers.

[1]Note that we do not require the knowledge of that mapping or, for that matter, any guidance to the nature of the function(s) mapping from external appearances to intrinsic behavioral traits.

In this paper, we adopt a binary interaction outcome scenario, where in each interaction, each party can choose to cooperate ($C$) or not ($\overline{C}$) with the opponent. We assume that each agent, both in the incoming and existing populations, have an external, visible feature set or tag and an internal hidden matching mechanism that determine whom that agent will cooperate with. Matching an agent implies cooperating with that agent, while not matching implies not cooperating. We investigate various scenarios of varying payoffs for matching and not matching, corresponding to different real world situations.

We evaluate several combinations of evolutionary and classical learning approaches to see if the incoming population can successfully develop effective interaction policies from repeated interactions. Learning of such policies is key for a successful assimilation of the incoming population into the stable one. Our results show varying degree of success with different representation scheme and dictated by the correlation between the external features and the intrinsic behavioral traits of the stable population.

In our study we also tried to create scenarios where learning agents are not best served by pure strategies like "always cooperate" or "always defect", but are better served by conditional policies like "cooperate with cooperators and defect against defectors". We propose a method to adjust the payoff matrix of the game such that the evolving agents are incentivized to play conditionally depending on its opponent's expected behavior rather than committing to defecting or cooperating unilaterally.

## 2. RELATED WORKS

Previous work on tag-based mechanisms of game-playing and cooperation have focused on developing evolutionary schemes to learn optimal actions based on tag relationships to strategies [12, 10, 11]. The proposed mechanisms are generally developed to be played in iterated stage games of Prisoner's Dilemma or the Anti-Cooperation game.

Such mechanisms invoke population models of learning, in which agents may develop identities via mimicry [10]. Matlock and Sen observe that while some games, such as the Anti-cooperation game, may discourage mimicry, other games like the Prisoner's Dilemma empower the tagging mechanism, and mimicry becomes a prominent strategy.

Such models provide effective mechanisms which consistently produce social rationality and/or Nash equilibria [6]. The Principle of Social Rationality, as proposed by Hogg and Jennings [7], is: *If a socially rational agent can perform an action whose joint benefit is greater than its joint loss, then it may select that action.* The goal of Hales' evolutionary model is to produce emergent dynamics which result in a system of socially rational agents. We are interested in producing socially rational agents, but stubborn agents in our system that refuse to cooperate force the learning agents in the system to not cooperate with others as well. As a result, the problem is one of classifying stubborn agents based on their outward attributes and cooperating with those that maximizes the number of cooperations, followed by non-cooperations.

Previous work on tag mechanisms also focuses on changing tag representation over the matching representation [5]. Our model is interested only in tags composed of bit strings, and we suggest four novel improvements to the matching mechanism that increase performance in the cooperation game.

Stereotyping in trust systems shares some very similar qualities with tag-based mechanisms, and similar techniques have been used to construct such stereotypes [2, 13]. These approaches use a machine learning classifier to separate agents into groups, with the intent of making generalizations about groups to determine the trustworthiness of agents inside such groups. Similar applications of classification in multi-agent frameworks focus on subjects such as text classification [4]. We are interested in developing similar classification schemes to map visible feature attributes into strategies.

Unlike the aforementioned papers, our approach uses an evolutionary scheme to generate a classifier. While previous papers have gathered information based on single interactions, we are interested in population-level interactions which is more realistic in social situations where interactions are numerous and short. The evolutionary mechanism we introduce includes sharing of tag information among the evolving population as well as strategic information. Finally, our agents play a different simultaneous game in which cooperation from both players is encouraged, and otherwise the preferred action is to not cooperate. This game models many realistic situations, in which agents are motivated to cooperate with others but because of preconceived negative biases, there are those who will not cooperate with certain groups. In this case, players are motivated to not attempt cooperation with these agents as well, as some sort of negative retribution. In some ways, players in this game are learning to be socially rational, but in contrast to previous works there exists a group of irrational agents (known as "static agents") which act irrationally in all respects.

## 3. MATCHING METHODS

In the current framework, a population of agents is divided into two groups: a static population and an evolving population. Each agent has two properties; a tag and a matching mechanism. Agents in the static population have fixed strategies and tags. The goal of the evolving population is to collaborate to develop a combination of tags and strategies that maximizes their payoffs when interacting with all the members of the static population. Each interaction between a newcomer and a member of the static population corresponds to a pre-defined stage game, the Coordination game.

The matching methods we present now considers the only "visible" features of their opponents, their tags, and do not have any other information, e.g., their individual identity, past performance, etc. In other words, incoming agents can see their opponents' tags, and their own tag and matching mechanism, while making the decision to cooperate or not. This way instead of recognizing each agent with their identity, incoming agents create matching mechanisms based on stereotypes, and base their strategy on their matching mechanism.

The use of tags in a population of agents has been shown to induce coordination [12]. Tags used in this research are binary strings of length $TL$, representing the observable binary attributes of an agent. Matching mechanisms are the methods used by the agents to decide either they will cooperate or not with an agent $a$ with associated tag $T_a$. In this study four different matching mechanisms are used:

**Ternary Matching Strings:** Agents use a ternary matching string $MS$ composed of values in $\{0, 1, *\}$, where $*$

corresponds to a don't care, and of length equal to the length of the agent tags. For an agent $a$ with a ternary matching string to cooperate with agent $b$,

$$M_{MS}(a,b) : \forall_{i \in \{1,2,\ldots,TS\}} MS_a(i) = T_b(i) \vee MS_a(i) = *$$

Where $MS_a(i)$ corresponds to value of the $i$th position of $a$'s matching string and $T_b(i)$ is the value of the $i$th position of the tag of agent $b$. So, a match occurs if, for every position, either the matching string contains the same value as the other agent's tag or contains a don't care symbol. Such a matching approach makes sense when agents decide on which tag positions or features are important, and judge others based on whether they meet the criteria on these salient features.

**Hamming Distance:** Agents decide to cooperate based on the similarity of their own tags with that of others. In Hamming matches, the total number of bit differences between the tags of two agents $a$ and $b$ must be at least $H_{min}$ but no more than $H_{max}$. In other words, agents should prefer those similar to them, but prefer at least some difference:

$$M_H(a,b) : H_{min} \leq |\{i \in \mathbb{Z} : TS_a(i) \neq TS_b(i)\}| \leq H_{max}$$

So matching is based only on the tags of the two interacting agents.

**Decision Tree:** In this matching mechanism, an agent uses a decision tree which is used to classify the opponent's tag to decide whether to cooperate or not. Nodes in the tree correspond to tests on tag features and results in one of two outcomes. Decision trees can compute arbitrary functions on boolean features and hence the range of behaviors that can be represented is greater than that can be represented with the matching mechanisms using Hamming distance or ternary strings.

**Intelligent Classifiers** While the above three matching mechanisms are used by the static population, for the new agents only we also incorporated learning: a new agent interacts with each static agent and the tags and cooperating decisions made by those agents are used as the training set for the learning algorithm. This approach uses the history of previous interactions for an agent to determine which combination of attributes leads to cooperation or defection. Unlike other approaches, the classifier method of matching solely uses the interaction history to decide future actions and newcomers do not have to share information with other newcomers. For learning the matching function from training data, we used a Random Tree classifier[1], though any other supervised classification algorithms including neural networks, Bayesian classifiers, Support Vector Machines, etc. can also be used.

We have performed a series of experiments with different population configurations using different matching mechanisms. In a given configuration, all static agents use the same matching mechanism, which can be different from the matching mechanism used by all members of the incoming population. All but the `Intelligent Classifiers` matching method has been used with the static population members (this is because the learning classifier will change the existing population from a static to a dynamic one). Similarly,

| | | | | |
|---|---|---|---|---|
| $M_i$ : | 0 | 1 | * | 1 |
| $M_{i'}$ : | 0 | 1 | 0 | 1 |
| $T_j$ : | 0 | 1 | 1 | 1 |

**Figure 1: An example of a ternary matching string and tag interaction. Player $i$ will cooperate with $j$, but $i'$ does not cooperate with $j$ due to a mismatch in values at index 3.**

| | | | | |
|---|---|---|---|---|
| $T_i$ : | 0 | 1 | 0 | 1 |
| $T_j$ : | 1 | 0 | 1 | 0 |
| $T_k$ : | 1 | 1 | 0 | 1 |

**Figure 2: The Hamming distance between $T_i$ and $T_j$ is 4, and between $T_i$ and $T_k$ is 1. For cooperation, agents need to be similar, but have some differences.**

all but the `Hamming Distance` and `Decision tree` matching method has been used with the incoming population members. This is because, as stated, the Hamming distance function is a fixed function and does not contain a learning opportunity (though there is a possibility for learning $H_{min}$ and $H_{max}$, which we have not explored in this paper).

## 4. EVOLVING THE NEW POPULATION

The members of the incoming population need to adapt their behavior and external appearance, their tags, so as to realize maximum utility from interaction with all the members of the static population. We use an evolutionary framework, where each newcomer interacts with each existing agent. Their individual experiences can be used, if desired to, to both adapt their personal strategies from their "local knowledge" or personal interaction histories with the static population, e.g., that done when the `Intelligent Matching` method is used[2], and also shared with other newcomers to identify the most preferred tag for the newcomers, by using an evolutionary algorithm that utilizes this "global knowledge". As this is an evolutionary mechanism, the agents do not keep track of all the interactions they have with each static agent. The only information they have is the utility they got in the end of a generation. They use this information to keep their behavior or adopt some other agent's behavior.

In the following we present the evolutionary process used to learn the tags and strategies of the incoming population.

For every generation, the population is able to reproduce asexually; new agents will be selected and have a possibility of mutating every generation, but there is no sexual recombination of tags or match data. The fitness of an agent in the evolving population is calculated by the cumulative payoff of games played against the entire static population. At each generation, tournament selection is used to decide the new population. Every agent in the existing population will participate in selection; these agents are given the choice of either mimicking the identity of a random agent or preserving their own. The final decision is made by selecting the agent with the highest fitness value of the two with the probability of $P_{Selection}$. This means the better agent will be selected with $P_{Selection}$ and the worse one still has the

[2]When the evolving population of newcomers use the learning classifiers on their personal interaction history, we have a hybrid evolutionary machine learner.

```
for g generations do
    for each agent, A_E, in evolving population, P_E do
        for each agent, A_S, in static population, P_S do
            if A_E.matches(A_S.tag) then
                if A_S cooperates then
                    A_E gets P_CC;
                else
                    A_E gets P_{\overline{C}C};
            else
                if A_S doesnotcooperate then
                    A_E gets P_{C\overline{C}};
                else
                    A_E gets P_{\overline{CC}};
        calculate the fitness, average payoff, of A_E;
    Using tournament selection generate new P_E;
    for each agent, A_E, in P_E do
        Mutate Tag and Match with probability μ;
```

**Algorithm 1:** Evolutionary Algorithm

chance to be selected with the probability of $1 - P_{Selection}$. After selection, every agent will be subject to a mutation stage given a parameter mutation probability $\mu$.

The mutation operation used in the evolutionary process is dependent on the type of matching mechanism used in the experiment. The Hamming distance method is not used in the evolutionary population, so no mutation operator is used for that matching mechanism. For ternary Matching String (MS) mechanism the mutation function traverses each bit of the MS and replaces it with one of 0,1 or * with the probability of $\mu$. Mutation is not applied to the matching function generated by the intelligent classifier.

## 4.1 Payoff Matrix Adjustments

In most of the cooperation problems, pure strategies, which are very easy for the agents to come up with, create a good enough utilities so the agents become unwilling to move from that strategy and search for better strategies. Dreżewski shows that these strategies would cause the agents to get stuck at a local maximum instead of searching for a global one [3]. In most real life scenarios, people rarely choose to use pure strategies (interact with or ignore everybody), and instead adopt selective strategies.

Research on Commodity Theory has shown that the scarcity of some item is inversely proportional to its value [9]. Since cooperation and non-cooperation are some commodity, the same principle may impact the development of stereotypes. For example, an interviewee may attempt interviews with more interviewers if hiring is low; however, when everyone is hiring, the interviewee would desire to not interview with those that may not need them. To that end, the payoff matrix is adjusted to take into account the cost of heterogenous outcomes (where the learning and static player choose different strategies) given the likelihood of cooperation for a static population: if the static population is likely to cooperate, then the cost of cooperating with non-cooperators increases; conversely, if the static population is not likely to cooperate, the cost of cooperating with non-cooperators increases. By calculating an 'unstabilized point', we want to achieve a society which will be willing to move away from pure strategies. That way the agents will be more likely to find the mixed strategies, which will achieve better utilities.

As mentioned above, each interaction between a newcomer and a current member of the static population corresponds to a stage game. The payoff matrix of the game incentivizes the evolving strategies and matching mechanisms of the incoming population. We now outline our design of the payoff matrix so that incentivizes the emergence of conditional matching strategies, that respond to the matching behavior of the opponents, rather than "always cooperate" or "always defect" matching behavior. For any random tag and a matching mechanism, the probability of matching is affected by the matching mechanism used and the parameters of the system. $\Delta$ signifies the difference in payoff of pure cooperation (that is, given a static population, the learning agent chooses to cooperate in all cases) and pure non-cooperation. We then estimate the value of $\Delta$ for a given system of static agents:

$$E[\Delta] = k(P_{CC} - P_{\overline{C}C}) + (1-k)(P_{C\overline{C}} - P_{\overline{CC}}) \qquad (1)$$

where $k$ is the probability that a member of a static population will cooperate given a random tag. $k$ is affected by the matching mechanism used by a static agent: the tag matching mechanism, for example, has a very low $k$ since one string yields very few cooperations across the set of possible tags.

There are three possible configurations of $\Delta$ which will radically effect the dynamics of the evolutionary learning system:

1. When $\Delta >> 0$, (is significantly greater than 0) a random learning population will be rewarded for the few agents they cooperate with; as such, learning populations will converge towards a total cooperate strategy;

2. When $\Delta << 0$ (is significantly less than 0) the learning population is rewarded for not cooperating, and will instead converge towards total non-cooperation;

3. The special case of $\Delta \approx 0$ represents when the payoff matrix does not incentivize the learning population towards either total cooperation or defection; then any naive mixed strategy that ignores tags would yield approximately the same payoff; agents then may be led to correctly classify the static population.

For an increase in payoff, learning agents must cooperate with cooperative static agents and correspondingly not cooperate with non-cooperative agents. Since the trivial solutions for $\Delta >> 0$ and $\Delta << 0$ may be easily developed, we further examine the case of $\Delta = 0$, and the payoff matrix in figure 4.1 is the solution to equation 1, by contriving the payoff values for cooperation and non-cooperation.

Finally, we calculate $k$ for every matching mechanism used for the static population. Consider $k_{TS}$, the probability of matching a random tag given a random ternary matching string:

$$k_{TS} = P_m^{TL}, \qquad (2)$$

where $P_m$ is the probability of matching one bit. For a ternary string this value is $\frac{2}{3}$ since, the possible match values $\{0, 1, *\}$ are equally probable and for any bit in $\{0, 1\}$ will match one item in $\{0, 1\}$ and will definitely match *.

| | $C$ | $\overline{C}$ |
|---|---|---|
| $C$ | $4$ | $\alpha$ |
| $\overline{C}$ | $P_{CC} - \frac{1-k}{k}(P_{\overline{CC}} - \alpha)$ | $2$ |

**Table 1: Payoff matrix for the evolving population (row player) against the static agent population (column player) for $\alpha = 1$ and probability $k$.**

The corresponding probability for decision tree matching, $k_{DT}$ is

$$k_{DT} = 0.5. \tag{3}$$

Consider the decision tree $D$ in the set of all possible decision trees for a given domain of tags. Then, a complement decision tree $D'$ may be constructed by switching the binary classification of every leaf node on the tree. So $D$ and $D'$ have opposite classifications, and this transformation is unique for $D$ and $D'$. Since every decision tree has such a complement, any decision tree that may cooperate with a given tag will have another tree that does not cooperate. Therefore, the probability of cooperation with a randomly generated decision tree for any tag is 0.5.

Finally, we calculate the probability of matching for Hamming distance, $k_{HD}$:

$$k_{HD} = \sum_{i=H_{min}}^{H_{max}} P_m^{TL-i} * (1 - P_m)^i * \binom{TL}{i} \tag{4}$$

When the agents in the static population uses Hamming distance as their matching mechanism, the probability of a random generated tag matching their mechanism can be calculated as in Equation 4. $P_m$ is the probability of matching a bit as it was in the definition of $k_{TS}$. $H_{min}$ is the minimum Hamming Distance at which the agent will cooperate, and $H_{max}$ is the corresponding maximum.

# 5. EXPERIMENTAL RESULTS

We now present results of our experiments with different population configurations, using different combinations of matching mechanisms.

Unless otherwise specified, simulations are run until convergence (identical tags and matching strings in the learning population) or until the system exceeds 200 generations (in which the system likely never converges). Our learning population has 100 agents, to be tested against a population of 1000 static agents. The mutation parameter $\mu = 0.001$; the probability of mutation is independent between tags and matches. The tag length of all agents is 8, and for Hamming matches the parameters have the value $H_{min} = 2$ and $H_{max} = 4$. The selection parameter is set to $P_{Selection} = 0.8$ in the setup.

Among all static populations, the degree of cooperation on average was affected by the matching mechanism. Levels of cooperation were dictated by the degree of discrimination in matching mechanisms for static populations (as in, higher percentages of cooperation were determined by the average number of tags that would be accepted by the matching mechanisms of the static population). Conversely, learning populations required intelligent discrimination to achieve mutual outcomes ($CC$and $\overline{CC}$); the evolutionary technique performed well in discriminatory techniques.

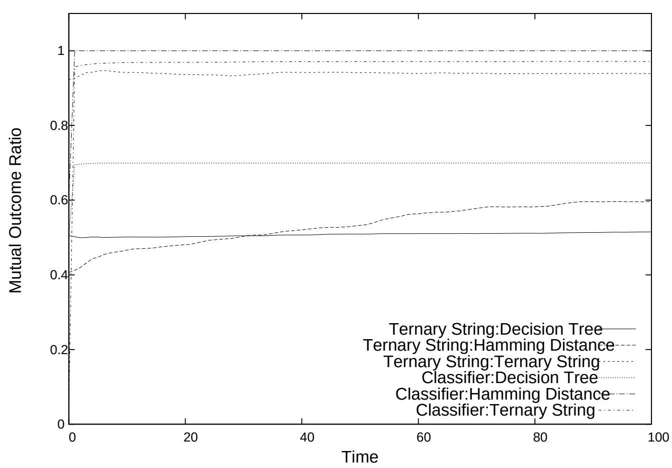


**Figure 3: Ratio of mutual outcomes to non-mutual outcomes over time averaged for 20 trials. (in format Learning:Static matching mechanism)**

Results shown in Table 3 represents maximal cooperation or non-cooperation that may be induced from the static population. Data is averaged over the same set static populations used in Table 2. The maximal percentage of cooperation is determined by finding the tag configuration which yields the highest number of cooperations for a given static population. A similar metric is used for finding the maximal percentage of non-cooperation.

Intelligent classifiers performed best in the total rate of mutual outcomes against all static population types. The intelligent classifier produces near-optimal matching since it is trained against the entire population of static agents. The effectiveness of agent cooperation, then, is measured in relation to their performance against the intelligent population.

Hamming distance was the least discriminatory of all matching mechanisms used in the static population. Unlike other matching mechanisms, it is the tag that determines the behavior of an agent using the Hamming distance mechanism; as such, agents with identical tags would share identical behaviors. As such, the intelligent classifier will always have total mutual outcomes as it has all the information necessary to predict the behavior of the static population.

Ternary string-based matching was the most discriminatory; on average, a learning agent could only cooperate with 5% of the static population. This pattern was accounted for in the adjusted payoff matrix. So the learning populations were unwilling to cooperate, and only did cooperate with a small number of static agents. Ternary strings are more discriminatory, so by their nature learning ternary strings will have high rates of non-cooperation.

Of the three static matching mechanisms, the decision tree has by far the most complex hidden strategy. Nonetheless, both learning populations were able to induce high levels of mutual outcomes. Ternary strings and the classifier has equal levels of mutual noncooperation, but the classifier population outperformed in mutual cooperation. The ternary string's innate discriminatory ability likely contributed to its high level of noncooperation; the fact that it could evolve equal levels of mutual cooperation reveals the benefit of applying the evolutionary framework to this problem.

| Learning Match | Static Match | Fitness | Mutual Outcomes | $CC$ | $\overline{CC}$ | $\overline{C}C$ | $C\overline{C}$ |
|---|---|---|---|---|---|---|---|
| Ternary | Ternary | 1.46 | 93.35 | 00.43 | 92.91 | 02.23 | 04.42 |
| Classifier | Ternary | 1.48 | 97.14 | 00.66 | 96.48 | 02.33 | 00.53 |
| Ternary | Hamming | 2.85 | 61.62 | 37.60 | 24.03 | 25.42 | 12.96 |
| Classifier | Hamming | 3.28 | 100.00 | 64.13 | 35.87 | 00.00 | 00.00 |
| Ternary | Decision | 2.49 | 53.35 | 25.55 | 27.80 | 26.04 | 20.61 |
| Classifier | Decision | 2.74 | 69.99 | 42.61 | 27.38 | 09.69 | 20.32 |

**Table 2: Percentage of outcomes for the row player (evolving players) and the column player (static players). Averaged over 20 trials. Average rate of mutual outcomes for pure strategy agents with best possible tag included for comparison.**

## 5.1 Learning rates

We compare the rates at which incoming populations learn their tags and matching mechanisms in figure 3. Incoming populations using ternary strings and classifier matching mechanisms converged fairly quickly against all agent types. Classifier agents converge quickly due to the small search space; since the only evolving aspect is the tag string with 256 possible values, it quickly finds the best possible configuration given the classification.

Ternary string agents face a greater challenge with a larger search space. The time for convergence is remarkably larger for ternary strings learning against Hamming distance. The challenge that ternary strings face is the high likelihood of structural noncooperation; for mechanisms like Hamming distance, given our parameters, agents will cooperate with approximately 60% probability. In response, the payoff matrix is adjusted to punish more for cooperation with noncooperators; ternary strings are able to construct an effective ”model agent” that yields high degrees of cooperation (since static agents act according to their tags) at the cost of ignoring many potential static cooperators. Interestingly, the ternary learning populations found a balance between cooperation and noncooperation; the longer convergence time is understandable given the rigid structure of ternary strings.

| Static Match | Pure $C$ | Pure $\overline{C}$ |
|---|---|---|
| Ternary | 5.56 | 97.57 |
| Hamming | 64.34 | 43.85 |
| Decision | 52.24 | 52.55 |

**Table 3: Percentage of outcomes for pure strategies of cooperation and non-cooperation, calculated on a trial-by-trial basis for each trial in Table 2.**

# 6. CONCLUDING REMARKS

Stereotyping is a core mechanism for identification and learning. Extracting hidden strategies from agents using judgments of external features is essential to functioning in a society. We present a set of automotive processes simulating the assimilation of agents into new cultures and synthesis of personal stereotyping mechanisms using a hybrid evolutionary and classifier approach. A set of matching mechanisms, which may classify agents into “cooperate” and “no cooperate” categories, are introduced to reproduce stereotype development. Agents cooperate in an evolutionary framework to learn inside an established society of agents.

Simulations consist of two populations: the evolving population and unchanging static population. Agents between both populations play a modified version of the Coordination game; the goal is for both players to play identical, or mutual strategies. Classifier agents using the Random tree classifier as their matching mechanism tend to perform the best when faced against such a problem;

In our current game the Classifier matching mechanism, although contrived, performs best out of the three learning schemes tested. Further work may involve changing the current game to punish classification schemes – which are not realistic and are non-cooperative responses to the stereotyping problem. For example, adding a cost to mutual cooperation which varies by agent would add the consideration of agent quality, instead of only quantity.

While the evolutionary mechanism in our results shows a remarkable ability to adapt to a wide variety of populations, modifications to the process may benefit the system further. Currently, the learning process is simultaneous over tag and match mechanism evolution: Separating these processes may produce mutual outcomes to a greater extent at the end of the simulation by allowing more convergence. Similarly, the inclusion of more evolutionary matching mechanisms, such as a decision tree, may expand the expressability of the evolutionary algorithm to address particularly complex static matching schemes.

# 7. REFERENCES


[1] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[2] C. Burnett, T. J. Norman, and K. Sycara. Bootstrapping trust evaluations through stereotypes. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1 - Volume 1*, AAMAS ’10, pages 241–248, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems.

[3] R. Dreżewski. A model of co-evolution in multi-agent system. In *Multi-Agent Systems and Applications III*, pages 314–323. Springer, 2003.

[4] Y. Fu, W. Ke, and J. Mostafa. Automated text classification using a multi-agent framework. In *Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, pages 157–158. ACM, 2005.

[5] D. Hales. Change your tags fast!–a necessary condition for cooperation? In *Multi-agent and multi-agent-based simulation*, pages 89–98. Springer, 2005.

[6] D. Hales and B. Edmonds. Evolving social rationality for mas using tags. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 497–503. ACM, 2003.

[7] L. M. Hogg and N. R. Jennings. Socially rational agents. In *Proc. AAAI Fall symposium on Socially Intelligent Agents*, pages 8–10, 1997.

[8] C. M. Judd and B. Park. Definition and assessment of accuracy in social stereotypes. *Psychological review*, 100(1):109, 1993.

[9] M. Lynn. Scarcity effects on value: A quantitative review of the commodity theory literature. *Psychology & Marketing*, 8(1):43–57, 1991.

[10] M. Matlock and S. Sen. Effective tag mechanisms for evolving coordination. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, AAMAS '07, pages 251:1–251:8, New York, NY, USA, 2007. ACM.

[11] A. McDonald and S. Sen. The success and failure of tag-mediated evolution of cooperation. In *Learning and Adaption in Multi-Agent Systems*, pages 155–164. Springer, 2006.

[12] R. L. Riolo. The effects and evolution of tag-mediated selection of partners in populations playing the iterated prisoner's dilemma.". In *Proc. of Inter. Conf. Genetic Algorithms (ICGA-97), Thomas B ack (ed.)*, pages 378–385, 1997.

[13] A. R. Wagner. Using stereotypes to understand one's interactive partner. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 1445–1446. International Foundation for Autonomous Agents and Multiagent Systems, 2010.