Appeared in the Proceedings of the AAAI Conference on Artificial Intelligence, Senior Member Track, Blue Skies category, Jan. 2015.

Steering Evolution Strategically:

Computational Game Theory and Opponent Exploitation for Treatment Planning, Drug Design, and Synthetic Biology

Tuomas Sandholm
Computer Science Department
Carnegie Mellon University

Abstract

Living organisms adapt to challenges through evolution and adaptation. This has proven to be a key difficulty in developing therapies, since the organisms develop resistance. I propose the wild idea of steering evolution/adaptation *strategically*—using computational game theory for (typically incomplete-information) multistage games and opponent exploitation techniques. A sequential contingency plan for steering is constructed computationally for the setting at hand. In the biological context, the opponent (*e.g.*, a disease) has a systematic handicap because it evolves myopically. This can be exploited by computing trapping strategies that cause the opponent to evolve into states where it can be handled effectively. Potential application classes include therapeutics at the population, individual, and molecular levels (drug design), as well as cell repurposing and synthetic biology.[1]

Brief description of the main ideas

Living organisms adapt to challenges through evolution and adaptation. These survival mechanisms have proven to be a key difficulty in developing therapies, since the challenged organisms develop resistance.

It would be desirable to be able to harness evolution and adaptation for therapeutic and technological goals. For example, through a sequence of appropriate manipulations, could we get a heterogeneous population of cancer cells to evolve to benign ones? Or, could we steer the evolution of the population to a state where we can destroy it? Could we evolve bacteria that eat toxins from the environment?

In this paper I propose the wild idea of steering evolution/adaptation *strategically*. I propose that this be done using computational game theory and opponent exploitation techniques. A sequential contingency plan for steering evolution is constructed computationally for the setting at hand.

For example, in the context of medical therapeutics, consider a setting where there is *treater* (*e.g.*, a doctor) treating a patient (*e.g.*, an individual or a population of people). The patient may have some *disease*, such as HIV, cancer, influenza, ebola, diabetes, schizophrenia, or obesity. The treater's task is to treat the patient over time. The treatments can include various drugs and cocktails of drugs, as well as other activities such as surgeries, exercise prescriptions, and so on. Treatments can also include gene interference and disruption therapies; nowadays it is even possible to turn off individual genes in a cell and to turn them back on. The treater may also be able to conduct tests. The disease can change over time. For example, the HIV virus mutates. The treatments affect how the disease changes over time.

Copyright © 2015, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

[1]Patent pending, 2012.

I propose that this be modeled as a (zero-sum) game between the treater and disease, with potentially both sequential and simultaneous moves. Then, solving the game model for Nash equilibrium (or its refinements) provides an optimal treatment plan assuming the disease plays rationally, that is, in the worst possible way for the treater. This is a safe approach. Another advantage is that it does not require a probabilistic model of the disease.

However, this may be overly conservative: the disease may not behave rationally. I propose that opponent exploitation techniques be used to take advantage of the disease's suboptimal play. Various approaches can be used for this, as I will discuss—ranging all the way to modeling the setting as a single-agent domain where the only player is the treater.

Furthermore, biological opponents have a distinct characteristic that we can exploit. **Evolution is myopic**: it does not look ahead in the game tree. That begets dramatic opponent exploitation opportunities. An example of this is evolving a disease into a **trap** where it can be easily attacked so that it is destroyed or becomes less powerful (*e.g.*, less virulent, less contagious, or less able to evolve in bad ways). More generally, the task is to compute a *strategy* for the treater in the (typically incomplete-information) game that causes the myopic opponent to obtain low utility.

Benefits

Algorithms can often solve games better than humans can—and in many cases optimally—so there is potential to generate better treatment plans than doctors and policy committees generate. Today's manual planning is rather *ad hoc* and unsophisticated from the perspective of the state of the art in game-solving algorithms—for instance in the ability to generate high-quality sequential plans. Most medical treatment today is myopic: the treater tries to take an action that improves the patient's health immediately. This puts the treater at the very same disadvantage that the opponent—evolution—has! In contrast, the treatment plans proposed in this paper may myopically make the patient worse in preparation for later, highly effective treatments. For example, the sequential treatment may cause a virus population in an individual to evolve into a population that can be effectively treated with medicines. Clearly some multistage treatment plans exist in practice already, such as radiosensitization—where a drug is administered that makes radiation more effective against a tumor—and chemosensitization. However, those plans are manually crafted, and they are much shorter and simpler than the plans contemplated here.

Because the proposed planning is automated, it is dramatically faster and requires fewer human resources. This means that custom plans can be generated for more specific population segments—even individuals.

The speed also enables the user of the system to con-

duct what-if analyses (sensitivity analysis) to test how the system-generated plan would change under different assumptions about the game, for example, structure of the game, probabilities of the player called *nature* that is used in game modeling to represent stochasticity (which can also be used to represent noise in test results), *etc*.

This has the potential to also guide future medical research. The most valuable knowledge to generate is knowledge that enhances the value of the treatment plans the most.

Applications

There are vast potential applications for these ideas. In this section I will briefly describe some application classes.

Battling disease within an individual patient

As an example of the use of this idea at the individual level, consider the treatment of a specific HIV patient. At each point in the game, the actions the treater can take may include treatments (such as which drug or drug cocktail to use) and tests. The actions the adversary (HIV) can take may include evolving the pool of the HIV viruses within the patient, making the patient worse or better in any number of ways, *etc*. There is data about the HIV virus as to what locations are the most typical mutation sites and what the most typical mutations at each site are. There is also data on treatment outcomes for different segments of the patient population.

In the game model, utilities can be associated with outcomes (leaves), intermediate states, and/or transitions. They can be based on the patient's health and projected health (including side effects), how virulent a state the disease is in, how contagious a state the disease is in, how easily attackable the disease is in its current state (*e.g.*, by a drug or cocktail), the cost of treatment and other costs to the patient/treater/insurance so far, projected future costs, *etc*.

The output is a strategy, that is, a contingent plan. It can be generated up front before treatment begins, or the planning can be interleaved with execution.

Battling disease at the molecular level: Drug design

As an example of the use of this idea at the molecular level, consider the design of drugs and drug cocktails for the treatment of an HIV patient. (Drug development to a specific individual may not be affordable, so this might be done for use on the entire HIV patient population or a segment thereof. However, the drugs can be used on an individualized basis as described in the previous section.)

The actions of HIV at any point in the game include the most likely mutations in the most likely mutating locations (binding sites) of HIV-1 Protease. (The normal operation of HIV-1 Protease is necessary for the propagation of HIV. The methodology can be used not only for protease, but also for the other typical targets in the HIV virus—reverse transcriptase and integrase—as well as other potential targets.)

The treater's available actions at any point in the game include which drug cocktail (what amounts of each drug) to use. Or, the actions can be to choose a cocktail of some existing drugs and some *de novo* drugs designed specifically by the present idea. In other words, the actions can include selection from a huge, or even infinite, space of potential drugs that have not even been conceived/manufactured before. It is not atypical for game models to be solvable even if the action space is infinite (Sandholm 2010). The actions of the treater can also include tests on the patient and the virus population in the patient.

A model can be used to predict how well each of the potential drugs in the potential cocktails would bind to each mutation at each site. Such models already exist (*e.g.*, (Kamichetty 2011; Langmead and Kamichetty 2011; 2012)) and will improve over time. The utility of the disease player can then be the sum over binding sites of the predicted binding energy at the site. The output is again a strategy.

Battling disease in a patient population

As an example of the use of this idea at the population level, consider battling the spread of an influenza epidemic. The actions of the influenza at any point in the game include spread of the various influenza strands—possibly including mutations—to different parts of the population. The actions of the treater at any point in the game include which drug or cocktail to use in each part of the population. (This is beyond the current way of treating influenza in the US where one cocktail is generated per year for the entire flu season, and the choice is merely whether to vaccinate a person or not.) The actions can also include quarantining, *etc*. In principle, the actions could also include the selection from an unrestricted space of drug design, if in the future drug manufacturing can be fast enough. The actions of the treater can also include conducting tests on patients from various parts of the population and testing aspects of the virus within such patients. The treater's utility in the game can be, for example, based on the number of deaths and other costs such as hospitalizations. The output is again a strategy.

The three levels of battle can also be combined in pairs, or one can combine all three. For instance, when battling a virus at the population level, one can include the molecular-level game. This can be especially helpful if the virus is new so there is little experience how it behaves in the population.

Cell repurposing and synthetic biology

The ideas also have applications beyond battling diseases. For example, one could apply them to repurposing cells. Could one evolve, say, a blood cell into a liver cell? Could one perhaps even grow a missing organ or limb?

The ideas apply beyond medical treatment as well, for example in synthetic biology. For instance, could one evolve bacteria into ones that eat toxins—such as oil spills–without introducing foreign genetic material?

Tackling questions in natural science

Beyond applications, the approach also enables one to formalize and potentially answer fundamental questions in natural science. For example, can a certain kind of cell be transformed into a certain other kind of cell using evolutionary pressures using a given set of manipulations? How much more power do multi-stage treatment plans offer? Does there exist a strategy (that uses only a given set of available manipulations and tests) that will destroy a given diverse (*e.g.*, cancer) cell population in a way that leaves no persistors? What is inherently impossible to achieve via evolution?

Operationalizing the ideas

Game representations

Figure 1 illustrates the most common representation of incomplete-information games, the *extensive form*. Each node is labeled with the name of the player whose turn it is to move. Stochasticity is represented by including a special player called *nature* who moves based on fixed probabilities rather than playing strategically. Note that the nature player is not the disease. Incomplete information is represented by information sets; the player whose turn it is to move at the information set does not know which node of the information set is the actual game state at that point. Simultaneous moves can be represented by drawing them as sequential but hiding the earlier mover's move from the later mover by bundling the resulting nodes in an information set.

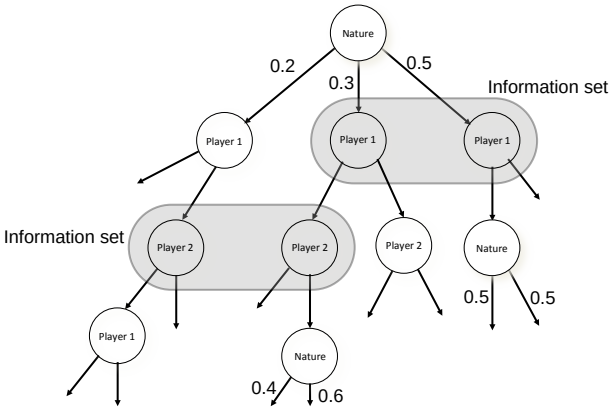

Figure 1: Small example of an extensive-form game.

An alternate representation In many biological applications, the opponent is a population rather than an individual. For example, a cancer typically consists of heterogeneous cancer cells and HIV can exist in multiple strands within a patient. This can cause the state of the opponent to be complex (high dimensional), leading to tractability issues.

The way I described the modeling so far, one would in this case model the state of the entire opponent population in a node of the game tree. The opponent (*i.e.*, population) gets to make one move at a time, as in traditional game theory.

An alternate, potentially more tractable, representation would be to allow the individuals in the opponent population to proceed down different paths of a tree or graph in parallel, *e.g.*, HIV could evolve into different strands (starting from one or more strands), each strand potentially having its own path (or each relatively small set of strands having its own path) in order to reduce the state dimensionality in each node. The population state would thus be represented not by one node but by a collection of nodes, potentially with population frequencies associated with the nodes. The notion of trap would then involve multiple traps in the graph: different strands could be trapped differently.[2]

[2]In principle, the setting could also be modeled as a game with one treater and multiple diseases (*e.g.*, each strand considered a separate disease), and the game would not have to be zero sum.

Data used to generate the game model The data for what moves the disease can make at different points in the game can be generated from scientific papers or databases of results, disease evolution models and simulations, tests on humans or animals (*e.g.*, modern *SMART trials* that test not just treatments but entire contingent plans for treatment (Shortreed et al. 2011)), past experience about the disease on a particular patient or segment of patients, experience gathered about the disease while using the system, active learning (Settles 2010; 2012), and so on.

The extra player, nature, can introduce stochasticity in the patient's state and in test results (probability of each reading conditional on the true state). The moves that nature can make, and the probability distribution over those moves, can be generated using the approaches described in the previous paragraph, potentially supplemented with data on the probabilistic errors that given tests have.

Computing game-theoretic equilibrium strategies

Game-theoretic solution concepts define how rational players should play a game. There are many solution concepts that can be used here. The most famous is Nash equilibrium, and several refinements thereof have been developed.

There are numerous algorithms for findings solutions to games according to a given solution concept. The scalability of algorithms for incomplete-information games has increased by orders of magnitude over the last ten years—largely driven by work on poker (for a review, see Sandholm (2010)). The leading approach involves running an abstraction algorithm to construct a smaller, strategically similar game, then computing an (approximate) equilibrium of the abstract game, and then mapping the computed strategies to the original game. For a recent review of abstraction in games, see Sandholm (2015). For algorithms for complete-information games, see, *e.g.*, Russell and Norvig (2010).

A desirable aspect of this game-theoretic approach is that it does not need much prior knowledge about the opponent (*e.g.*, detailed statistical medical knowledge of the disease). It amounts to assuming that the opponent behaves in the worst possible way for us. This is safe in the two-player zero-sum case: if the opponent actually does not behave in this way, that can only help us.

Opponent modeling and exploitation

Assuming a worst-case opponent can be overly conservative in settings where we know (or believe) that the opponent will not behave rationally. Here, opponent modeling and opponent exploitation can be highly beneficial.

In essence, an opponent model predicts what the opponent would do—perhaps probabilistically—in various information sets. In our context, opponent models can be generated from scientific papers or databases of results, disease evolution models (*e.g.*, (Frey et al. 2010)) and simulations, tests on humans or animals (*e.g.*, SMART trials that test entire contingent plans for treatment (Shortreed et al. 2011)), past experience about the adversary, experience gathered about the adversary while using the system, active learning (Settles 2010; 2012), and so on. For example, to generate the

opponent model of HIV, one could use data on which antivirals tend to cause specific mutations in reverse transcriptase, protease or integrase (*e.g.*, in the form of a probability table), and data on efficacy of other antivirals against such mutants.

Then, there are many approaches to opponent exploitation (*e.g.*, (Sandholm 2010; McCracken and Bowling 2004; Billings et al. 2004; Bard and Bowling 2007; Southey et al. 2005; Hoehn et al. 2005; Sturtevant, Zinkevich, and Bowling 2006)), and additional ones are being developed each year.

As one example, one can start by playing game theoretically and then adjust play toward exploiting the opponent in points of the game that have been frequently visited so there is good statistical information about the opponent's play at those points (Ganzfried and Sandholm 2011).

As a second example, one can compute an ϵ-*safe best response* (or approximation thereof), *i.e.*, a strategy that exploits our model of the opponent maximally subject to the constraint that even against a worst-case opponent it will do at most ϵ worse than a game-theoretic strategy (Johanson, Zinkevich, and Bowling 2007; Johanson and Bowling 2009).

As a third example, one can compute a set of strategies and then use learning (in a simulation or in the real world) to determine which one of the strategies performs best against the actual opponent. *No-regret learning* algorithms are one natural way of conducting this learning (Bard et al. 2013).

As a fourth example, if one trusts the opponent model enough, one can abandon game-theoretic safety completely and compute a best-response strategy (or an approximation thereof) to the opponent model. This can be computationally complex in large games and with lots of randomness in the game. To find solutions for this setting, techniques from stochastic optimization can be leveraged, such as *trajectory-based optimization* (*e.g.*, based on sample trajectories of possible futures) and *policy gradient techniques*.[3]

At the other extreme, one could assume no prior knowledge of the opponent and yet require that one's opponent exploitation performs at least as well in expectation as a game-theoretic equilibrium strategy. Perhaps surprisingly, it turns out that it is indeed possible to exploit an opponent more than any game-theoretic equilibrium strategy can, while still having this safety property (Ganzfried and Sandholm 2014). Intuitively, if we can measure—or at least bound from below—how much value the opponent has gifted to us through suboptimal moves, we can use that value to bankroll our risky exploitation while guaranteeing safety overall.

Biological opponents have a distinct characteristic that we can further exploit. **Evolution is myopic**: it does not look ahead in the game tree. That begets dramatic opponent exploitation opportunities. An example of this is a **trap**. More generally, the task is to compute a strategy for ourselves that yields low utility to the opponent.[4]

[3] An interesting approach to Markov Decision Processes under certain forms of risk aversion involves converting the problem into a zero-sum game and solving it (Chen and Bowling 2012). The domain on which that algorithm was tested was a simplified diabetes management setting. That work is very different than what is proposed here.

[4] This can be generalized to opponents with limited lookahead deeper than one. I do not (yet) see biological applications for that.

Related work, briefly

There is an interesting piece of work proposing a game-theoretic approach to drug cocktail selection at the molecular level (Kamichetty 2011; Langmead and Kamichetty 2011; 2012). Their approach differs significantly from ours. First, the players' choices for the game are actions, not contingent plans. The equilibrium lives in the space of stable endpoints of a simulation rather than in the space of strategy profiles where the strategies are contingent plans. That makes the game single shot, has implications on the solving methodology, and restricts the power and generality. In contrast, the present paper considers multi-stage games, where traps and opponent exploitation are particularly powerful—and one can capture information-gathering actions (such as tests) and game-theoretic screening devices in the model. Second, their players are not a disease and a treater. In their HIV analysis there is one player for each likely mutation site, and each of those players has the likely mutations as its possible actions; there is also one player whose action space is the proportions of drugs from a set of three FDA-approved drugs to administer. In their other analysis (on PDZ), there are 21 players corresponding to variable positions (16 protein players and 5 drug players); each protein player has five actions corresponding to the wild-type and the four other most likely amino acid positions at this position. In these two ways that work is similar to Pérez-Breva et al. (2006).

There has been significant work on *evolutionary game theory* (*e.g.*, (Maynard Smith 1982; Axelrod 1984; Weibull 1995)). That research has typically studied competition among species rather than treatment. Also, the solutions—typically so-called *evolutionarily stable strategies* in a simple repeated game model—have usually been derived by hand rather than requiring sophisticated algorithms. Two interesting recent papers exist on game theory for cancer treatment—using two given drugs each (Basanta, Gatenby, and Anderson 2012; Orlando, Gatenby, and Brown 2012). The models were simple enough to solve analytically, and the strategies were only evaluated in simulation.

There has been significant work on *understanding disease evolution* (*e.g.*, (Ewald 1994)). In contrast, this paper is about steering evolution and methods for computing sequential (treatment) strategies. Also, the prior work has typically not taken a game-theoretic or optimization approach.

What needs to be done to realize this vision?

Significant work lies ahead to realize this sweeping vision. The game model needs to be instantiated on a real disease to test the ideas. New game representations may be needed to make the modeling and planning tractable. One may also need to develop tractable game solving and/or opponent exploitation algorithms if it turns out that existing ones do not scale. This may involve custom game abstraction techniques and possibly custom equilibrium-finding algorithms.

One also needs to test the efficacy of the plans in the wet lab. Fortunately, there exist tens of inexpensive cancer cell lines for doing so. Similar tests can also be conducted on viruses, on cell evolution for synthetic biology, *etc*.

References

Axelrod, R. 1984. *The Evolution of Cooperation*. Basic Books.

Bard, N., and Bowling, M. 2007. Particle filtering for dynamic agent modelling in simplified poker. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 515–521.

Bard, N.; Johanson, M.; Burch, N.; and Bowling, M. 2013. Online implicit agent modelling. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.

Basanta, D.; Gatenby, R.; and Anderson, A. 2012. Exploiting evolution to treat drug resistance: Combination therapy and the double bind. *Molecular Pharmaceutics* 914–921.

Billings, D.; Bowling, M.; Burch, N.; Davidson, A.; Holte, R.; Schaeffer, J.; Schauenberg, T.; and Szafron, D. 2004. Game tree search with adaptation in stochastic imperfect information games. In *Proceedings of the 4th International Conference on Computers and Games (CG)*, 21–34. Ramat-Gan, Israel: Springer-Verlag.

Chen, K., and Bowling, M. 2012. Tractable objectives for robust policy optimization. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*.

Ewald, P. 1994. *Evolution of Infectious Disease*. Oxford University Press.

Frey, K.; Georgiev, I.; Donald, B.; and Anderson, A. 2010. Predicting resistance mutations using protein design algorithms. *Proceedings of the National Academy of Sciences* 107(31):13707–13712.

Ganzfried, S., and Sandholm, T. 2011. Game theory-based opponent modeling in large imperfect-information games. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.

Ganzfried, S., and Sandholm, T. 2014. Safe opponent exploitation. *ACM Transaction on Economics and Computation (TEAC)*. Best of EC-12 special issue.

Hoehn, B.; Southey, F.; Holte, R. C.; and Bulitko, V. 2005. Effective short-term opponent exploitation in simplified poker. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 783–788.

Johanson, M., and Bowling, M. 2009. Data biased robust counter strategies. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Johanson, M.; Zinkevich, M.; and Bowling, M. 2007. Computing robust counter-strategies. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*.

Kamichetty, H. 2011. Application: Games of molecular conflict. In *Structured Probabilistic Models of Proteins across Spatial and Fitness Landscapes*, Carnegie Mellon University, School of Computer Science PhD thesis (CMU-CS-11-116). chapter 9, 121–127.

Langmead, C., and Kamichetty, H. 2011. Resistance-proof drug design. Provisional US patent application 61/464149.

Langmead, C., and Kamichetty, H. 2012. Using game theory in identifying compounds that bind to targets. PCT patent application PCT/US2012/026966.

Maynard Smith, J. 1982. *Evolution and the Theory of Games*. Cambridge University Press.

McCracken, P., and Bowling, M. 2004. Safe strategies for agent modelling in games. In *AAAI Fall Symposium on Artificial Multi-agent Learning*.

Orlando, P.; Gatenby, R.; and Brown, J. 2012. Cancer treatment as a game: integrating evolutionary game theory into the optimal control of chemotherapy. *Physical Biology* 9.

Pérez-Breva, L.; Ortiz, L.; Yeang, C.-H.; and Jaakkola, T. 2006. Game theoretic algorithms for protein-DNA binding. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*.

Russell, S., and Norvig, P. 2010. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3rd edition.

Sandholm, T. 2010. The state of solving large incomplete-information games, and application to poker. *AI Magazine* 13–32. Special issue on Algorithmic Game Theory.

Sandholm, T. 2015. Abstraction for solving large incomplete-information games. In *AAAI Conference on Artificial Intelligence (AAAI)*. Senior Member Track.

Settles, B. 2010. Active learning literature survey. Technical Report Computer Sciences Technical Report 1648, updated on January 26, 2010, University of Wisconsin–Madison.

Settles, B. 2012. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan Claypool.

Shortreed, S.; Laber, E.; Lizotte, D.; Stroup, T. S.; Pineau, J.; and Murphy, S. 2011. Informing sequential clinical decision-making through reinforcement learning: an empirical study. *Machine Learning* 84:109–136.

Southey, F.; Bowling, M.; Larson, B.; Piccione, C.; Burch, N.; Billings, D.; and Rayner, C. 2005. Bayes' bluff: Opponent modelling in poker. In *Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, 550–558.

Sturtevant, N.; Zinkevich, M.; and Bowling, M. 2006. Prob-max^n: Opponent modeling in n-player games. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 1057–1063.

Weibull, J. 1995. *Evolutionary Game Theory*. MIT Press.