Fitness among Competitive Agents: A Brief Note

By William A. Dembski

The upshot of the No Free Lunch theorems is that averaged over all fitness functions, evolutionary computation does no better than blind search (see Dembski 2002, ch 4 as well as Dembski 2005 for an overview). But this raises a question: How does evolutionary computation obtain its power since, clearly, it is capable of doing better than blind search? One approach is to limit the fitness functions (see Igel and Toussaint 2001). Another, illustrated in David Fogel's work on automated checker and chess playing (see, for instance, Chellapilla and Fogel 1999 and Fogel et al. 2004) and, more recently, given a theoretical underpinning by David Wolpert and William Macready (2005), is to limit optimization problems to search spaces consisting of agents that play competitively against one another. In this brief note, I focus on the latter attempt to get around the force of No Free Lunch.

To represent the problem, consider a large space of agents **Ω** with *N* elements. The elements of **Ω** are agents that play a game with a fixed set of possible payoffs: $s_1 < s_2 < \cdots < s_m$. We assume that for each *i* and *j* in **Ω**, if s_k is the payoff for *i* playing a particular game with *j*, then $s_m - s_k$ is the payoff for *j* in that same game. Since most games are either win-lose or win-lose-draw, *m* usually equals 2 or 3 so that s_1 equals 0 and s_2 equals 1 or s_1 equals 0, s_2 equals ½, and s_3 equals 1.

To make sense of No Free Lunch in such competitive environments now requires giving meaning to fitness in these environments inasmuch as NFL is inherently about averaged fitness. Suppose agent *i* obtains a bigger payoff than agent *j* when they play each other on a given occasion. Should the fitness of *i* therefore be bigger than *j*? Clearly not since on repeated play between them, *j* might on average score better than *i*. Minimally, then, fitness among such agents will require probabilities p_{ijk}, where this number denotes the probability that *i* receives payoff s_k in playing *j*. Note that summing these p_{ijk} over *k* equals 1.

If we now define

$$\gamma_{ij} = \sum_{k=1}^{m} s_k p_{ijk} ,$$

it would appear that this number could define a relative fitness of *i* in relation to *j*. Yet, this is not quite right. The problem is that in competitive environments, there can be intransitivities, in which one agent is pairwise superior to another and this other agent is pairwise superior to still a third, but the third is in fact pairwise superior to the first. Consider, for instance, the game Rock-Paper-Scissors. Each object in this game ties itself, defeats another, and is defeated by still a third: rock smashes scissors, scissors cut paper, and paper wraps rock (see http://www.worldrps.com).

A game matrix for Rock-Paper-Scissors is as follows:

	R	P	S
R	½	0	1
P	1	½	0
S	0	1	½

From the symmetry properties of this matrix, it is evident that just because one item happens to be pairwise superior to another does not mean that it is globally superior to the other. But that's precisely the challenge of assigning fitness of competitive agents inasmuch as fitness is a global measure of adaptedness to an environment.

To provide such a global measure of adaptedness and thereby to overcome the intransitivities inherent in pairwise comparisons, fitness in competitive environments needs therefore to factor in average performance of agents as they compete across the board with other agents. The simplest way to do this is to define the fitness of agent *i* as

$$\gamma_i = \sum_{j=1}^{N} \gamma_{ij} = \sum_{j=1}^{N} \sum_{k=1}^{m} s_k p_{ijk} \ .$$

Note that when m equals 2, s_1 equals 0, and s_2 equals 1, we can define p_{ij} as p_{ij2} (i.e., the probability of *i* scoring a 1 against *j* or, alternatively, the probability of *i* defeating *j*), and so

$$\gamma_i = \sum_{j=1}^{N} \gamma_{ij} = \sum_{j=1}^{N} p_{ij} \ .$$

That this is the proper way to define fitness in competitive environments is clear from considering how tournaments are decided when all the competitive agents are able to play each other (typically this is not possible because N, the number of agents in the space $\mathbf{\Omega}$, is too large for all agents to be paired in match play). Consider, for instance, a chess tournament like the St. Petersburg Chess Congress of 1909. The outcome of such a tournament is usually displayed in a tournament matrix as follows (taken from http://www.endgame.nl/stpeter.htm):

St. Petersburg 1909

		1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9		
1	Rubinstein	*	1	1	1	½	½	½	1	1	1	½	1	0	1	½	1	1	1	1	14½	875 Rubles
2	Lasker	0	*	½	1	½	1	1	1	½	1	1	1	0	1	1	1	1	1	1	14½	875 Rubles
3	Spielmann	0	½	*	1	0	1	1	½	1	½	½	½	1	0	½	1	½	½	1	11	475 Rubles
4	Duras	0	0	0	*	0	1	½	0	½	1	0	1	1	1	1	1	1	1	1	11	475 Rubles
5	Bernstein	½	½	1	1	*	0	1	0	1	1	1	1	½	0	0	0	½	½	1	10½	190 Rubles
6	Teichmann	½	0	0	0	1	*	0	½	½	½	½	1	1	½	1	½	1	1	½	10	120 Rubles
7	Perlis	½	0	0	½	0	1	*	½	½	1	½	1	1	½	1	½	0	0	1	9½	80 Rubles
8	Cohn	0	0	½	1	1	½	½	*	0	0	1	½	½	0	½	½	½	1	1	9	40 Rubles
9	Schlechter	0	½	0	½	0	½	½	1	*	1	0	0	1	1	½	0	1	½	1	9	40 Rubles
10	Salwe	0	0	½	0	0	½	0	1	0	*	½	1	1	1	½	0	1	1	1	9	40 Rubles
11	Tartakower	½	0	½	1	0	½	½	0	1	½	*	0	0	0	½	1	1	1	½	8½	
12	Mieses	0	0	½	0	0	0	0	½	1	0	1	*	½	1	1	1	0	1	1	8½	
13	Dus Chotimirsky	1	1	0	0	½	0	0	½	0	0	1	½	*	½	½	½	1	0	1	8	
14	Forgács	0	0	1	0	1	½	½	1	0	0	1	0	½	*	½	½	½	0	½	7½	
15	Burn	½	0	½	0	1	0	0	½	½	½	½	0	½	½	*	1	½	½	0	7	
16	Vidmar	0	0	0	0	1	½	½	½	1	1	0	0	½	½	0	*	½	1	0	7	
17	Speyer	0	0	½	0	½	0	1	½	0	0	0	1	0	½	½	½	*	½	½	6	
18	Von Freyman	0	0	½	0	½	0	1	0	½	0	0	0	1	1	½	0	½	*	0	5½	
19	Znosko Borovsky	0	0	0	0	0	½	0	0	0	0	½	0	0	½	1	1	½	1	*	5	

Emanuel Lasker, who tied with Akiba Rubinstein for first place, was at the time the world champion. Moreover, Rubinstein, though he never played Lasker in a title match (back then challengers had to raise sufficient funds before they could arrange a title match with the world champion), was in the five years preceding World War I regarded as the strongest player in the world (in 1912 he won five international tournaments in a row, a feat unparalleled). Yet both Rubinstein and Lasker were defeated by Dus Chotmirsky, a chess player who would be lost to history except for this feat.

A tournament, then, assigns fitness in terms of best overall performance when pairing up all agents in match play, with best overall performance measured as best sum-total of payoffs. Thus, for a tournament matrix $\tau = [\tau_{ij}]$ where *i* and *j* range over all the agents/players in Ω and where each of the τ_{ij} is the actual payoff for *i* from playing *j* in the tournament, one ranks players so that the player *i* for which the number

$$\tau_i = \sum_{j=1}^{N} \tau_{ij}$$

is greatest is assigned to the first row, the player *j* for which τ_j is the next largest is assigned to the second row and so on (with the order among ties being a matter of indifference). Indeed, this is exactly what is done with actual tournaments. It was done at St. Petersburg, as is evident in how cash prizes (in terms of Rubles) were allotted.

The tournament matrix $\tau = [\tau_{ij}]$ is the standard instrument for ranking winners on a given competitive occasion. Nonetheless, it leaves something to be desired as a reliable way of identifying the “most fit” agent(s)/player(s) in Ω. The problem is that a given player, who is otherwise excellent, might have been having a bad day. What’s needed, then, to assess fitness is not a single tournament matrix corresponding to a single tournament on a given competitive occasion, but rather an average of such tournament matrices over multiple trials so that lucky wins and unlucky breaks get factored out. In other words, the proper way to assess fitness is not by looking at a tournament matrix $\tau = [\tau_{ij}]$ from a single competitive occasion but rather by looking at the game matrix $\gamma = [\gamma_{ij}]$ that averages such tournament matrices. Given such a game matrix, fitness of a player *i* is given by

$$f(i) = \gamma_i = \sum_{j=1}^{N} \gamma_{ij} \ .$$

This measure of fitness corresponds precisely to how evolutionary computation seeks to find optimal competitive agents. Take David Fogel's work on evolutionary searches for optimal neural-net chess playing programs (see Fogel et al. 2004). He starts with some fixed but manageable number of randomly chosen chess playing neural nets (call this number *K*). Next he plays them against each other in all possible pairwise combinations. But he doesn't do it just once. He does it several times, thereby making all the neural nets play several tournaments among themselves (we assume this number is also fixed; call it *L*). Those neural nets whose average tournament performance falls below the 50th percentile get dropped in the next round of tournaments. Those that exceed the 50th percentile not only stay on but also reproduce (as is typical with evolutionary computation) so that the total number of neural nets competing in the next round of tournaments is again *K*. These, then, compete in another set of *L* tournaments, after which the cycle repeats with more rounds of tournaments, the number of rounds being constrained by computational resources. Wolpert and Macready (2005) generalize this set-up but don't add anything fundamentally new to it.

From this synopsis, it's evident that if *K*, the number of neural nets that compete in a given round of tournaments, could be enlarged to *N* and if *L* could be made arbitrarily large so that empirically calculated averages of payoffs converged to expected payoffs, then Fogel's method of finding optimal competitive agents would amount exactly to finding the largest fitness value γ_i associated with the game matrix $\gamma = [\gamma_{ij}]$. Only because of computational constraints does he limit the number of players and tournaments in a given round. Without such constraints, optimization among competitive agents would correspond precisely to a straightforward optimization of the fitness function $f(i) = \gamma_i$, with these values determined directly from the game matrix $[\gamma_{ij}]$. Fogel's approach is essentially to provide local approximations to the game matrix and then optimize with respect to those local approximations.

The question now remains whether fitness among competitive agents as determined by the game matrix allows for a "free lunch." It does not. The game matrix itself is determined by probabilities p_{ijk} and payoffs s_k where, recall, p_{ijk} denotes the probability that *i* receives payoff s_k in playing *j*. I leave it as an exercise to show that averaged over all possible p_{ijk} and s_k, competitive searches based on the game matrix do not outperform blind search (for simplicity assume that the s_k are real numbers between 0 and 1). Accordingly, NFL holds even for competitive agents.

And this brings us back to our first, and now final, question, namely, how does evolutionary computation obtain its power inasmuch as it is capable of doing better than blind search? So long as NFL applies, the fundamental intuition underlying evolution, namely, that an evolutionary process is able to end up with more than it started with, is hard to sustain. That's why David Fogel as well as David Wolpert look to competitive environments in which NFL supposedly breaks down. As Fogel put it to me in correspondence (1 June 2001), "Our experiment was about learning what would emerge from the ingredients we put in, trying to put in comparatively little knowledge about the game." The implication here is plain, namely, he views his chess and checker programs as outputting more knowledge than he put into them.

The view that evolution is a free lunch has been stated most forcefully by Richard Dawkins (1987, 316): "The one thing that makes evolution such a neat theory is that it explains how organized complexity can arise out of primeval simplicity." Getting organized complexity out of primeval simplicity is a good trick indeed. But it is a trick that NFL seems to debunk. The question therefore remains: Insofar as evolutionary computation does better than blind search, what is its source of power? That's a topic for another paper.

References

Chellapilla, Kumar and David B. Fogel (1999) "Evolution, Neural Networks, Games, and Intelligence," *Proceedings of the IEEE*, September, 1471–1496.

Dawkins, Richard (1987) *The Blind Watchmaker* (New York: Norton).

Dembski, William A. (2002) *No Free Lunch: Why Specified Complexity Cannot Be Purchased Without Intelligence* (Lanham, Md.: Rowman and Littlefield).

Dembski, William A. (2005) "Searching Large Spaces: Displacement and the No Free Lunch Regress," available at http://www.designinference.com.

Fogel, David B., Timothy J. Hays, Sarah L. Hahn, and James Quon (2004) "A Self-Learning Evolutionary Chess Program," *Proceedings of the IEEE* 92(12): 1947–1954, available online at http://www.natural-selection.com/Library/2004/92jproc12-fogel.pdf.

Igel, Christian and Marc Toussaint (2001) "On Classes of Functions for Which No Free Lunch Results Hold," available at http://www.no-free-lunch.org/IgTo01.pdf.

Lasker, Emanuel (1971) *The International Chess Congress, St. Petersburg, 1909* (New York: Dover).

Wolpert, David H. and William G. Macready (2005) "Coevolutionary Free Lunches," typescript, available through DHW at dhw@email.arc.nasa.gov.