Convergence of Simulation-Based Policy Iteration

In *Convergence of Simulation-Based Policy Iteration* (SBPI), Cooper, Henderson and Lewis are able to derive conditions under which a novel algorithm for computing optimal policies for Markov decision processes (MDPs). They point out that SBPI, originally suggested by Bertsekas and further developed by Cao, works similarly to the modified policy iteration algorithm (MPIA) and to the actor critic algorithm. It is therefore expected to converge faster than by the value iteration algorithm, yet the exact conditions under which SBPI converges had not been developed. That places the results of this paper in a position of great importance, and its applicability is shown in the description of methods to converge to estimators via simulation. These estimators can then be used to ensure the "almost-sure" convergence of SBPI.

Let us begin by comparing MPIA and SBPI. It is important to point out that the model used in this paper is the *average-reward* model, not the *discounted reward* model as studied thus far in IOE 512. This distinction required that I understand new measures:

- The *gain*, $g_\psi(x) \equiv \liminf_{k\to\infty} J_\psi^k(x)/k$, or the long-run average reward
- The *bias*, $h_\psi(x) = \sum_{n=0}^{\infty} E_\psi^x\left[r(X_n,\psi(X_n)) - g_\psi(X_n)\right]$

for a policy ψ given that the system started in state x. Puterman (p. 338) defines the bias as "the expected total difference between the reward and the stationary reward." These two definitions are key to understanding SBPI, whose algorithm proceeds as follows:

1. (Initialization) Choose a sequence $\{n_j, j \geq 0\}$ and a decision rule d_0. Let $j = 0$.
2. (Policy Evaluation Approximation) Obtain an estimate of the solution to the AEE $\left(g_{d_j}^{n_j}, h_{d_j}^{n_j}\right)$ for the decision rule d_j.
3. (Policy Improvement) Using our current estimate, find a decision rule d_{j+1} that satisfies $d_{j+1}(x) \in \arg\max_{a\in A_x}\left\{r(x,a) + \sum_{y\in X} p(y\mid x,a)h_{d_j}^{n_j}(y)\right\}$ for each $x \in X$, setting $d_{j+1}(x) = d_j(x)$ whenever possible.
4. (Iteration) Let $j = j + 1$ and return to step 2.

Since the AEE (*average evaluation equations*) are satisfied by a unique vector h, if we can find a solution (g, h) that satisfies $h = \max_{d\in D}\{r_d - g1 + P_d h\}$ the *average optimality equations* (AOE), then $g = g^*$ and we have found the set of optimal policies $D^* \equiv \arg\max_{d\in D}\{r_d + P_d h\}$. The SBPI algorithm relies on the fact that "all that is required for the policy improvement step is $h_{d_j}^{n_j}$." This allows $h_{d_j}^{n_j}$ to be estimated at each iteration through simulation, avoiding the inhibitive computations in the policy evaluation step of the policy iteration algorithm (PIA). In this sense, SBPI resembles MPIA, which can be described (in the average cost model) as follows (taken from Puterman, p.386):

1. Select $v^0 \in V$, specify $\varepsilon > 0$, and set $n = 0$.

2. (Policy Improvement) Choose d_{n+1} to satisfy $d_{n+1} \in \arg\max_{d \in D} \{r_d + P_d v^n\}$, setting $d_{n+1} = d_n$ if possible.
3. (Partial Policy Evaluation)
   a. Set $k = 0$, and $u_n^0 = r_{d_{n+1}} + P_{d_{n+1}} v^n$.
   b. If $sp(u_n^0 - v^n) < \varepsilon$, go to step 4. Otherwise, to to (c).
   c. If $k = m_n$, go to (e). Otherwise, compute u_n^{k+1} by $u_n^{k+1} = r_{d_{n+1}} + P_{d_{n+1}} u_n^k \equiv L_{d_{n+1}} u_n^k$.
   d. Increment k by 1 and return to (c).
   e. Set $v^{n+1} = u_n^{mn}$ and go to step 2.
4. Choose $d_\varepsilon \in \arg\max_{d \in D} \{r_d + P_d v^n\}$ and stop.

The two algorithms are similar in that they produce estimates ($h_{d_j}^{n_j}$ in the case of SBPI and v^{n+1} in the case of MPIA) that are used in the evaluation (SBPI) and improvement (MPIA) steps, respectively. Moreover, both have as stopping rules either a fixed number of approximations or ε-optimality, or both. I appreciated the comment in *Convergence* that "we leave the analysis of stopping rules as a subject for future research." Indeed, the paper does not explicitly give stopping rules, but only shows that in order to ensure convergence, under specified conditions, that one must run simulations *at least* a certain number of iterations (Theorem 4.2). The beauty of this result is its generality. Perhaps a stopping rule could appear in the form of a minimum threshold for the number of iterations accompanied by the condition that $d_{j+1} = d_j$.

For simplicity's sake, I appreciated the Ratio Estimator given in section 5.3. As the authors describe it, it "is perhaps the least complex to implement, as it can be applied based on a single simulated sample path." The estimator of $h(x)$, $h^n(x) = \frac{\overline{W}_n(x)}{\overline{C}_n(x)}$, where $\overline{W}(x)$ represents the mean "cumulative centered cost until the end of the current regenerative cycle when the chain hits x^*" given that the cycle began in state x, and where $\overline{C}(x)$ represents the mean number of visits to state x per regenerative cycle. Proposition 5.9 claims that $h^n(x)$ as defined above is indeed an unbiased estimator for $h(x)$, and that the sufficient conditions are met by virtue of the run lengths growing fast enough (ref. Proposition 5.5). Then, using the Ratio Estimator (or either of the other two estimators mentioned in the paper), convergence is ensured for the bias and hence for the policy. Again, however, a stopping rule for computing the $h^n(x)$ is not explicitly stated, leaving an open door for future research.

What I like about the paper, and the Ratio Estimator in particular, is that it shows how a computationally inhibitive MDP can be solved iteratively, in a finite (although perhaps very large) number of steps. It is possible to code such a problem, and the assurance of convergence outlined herein makes SBPI an attractive alternative. What I would like to know is how the convergence rate of SBPI compares with those of the actor-critic and MPI algorithms mentioned by the authors, as well as what constitutes a legitimate stopping rule in the computation of the $h_{d_j}^{n_j}$.