AIRCURRENTS: AN ADVANCED APPROACH TO INSURANCE PORTFOLIO OPTIMIZATION USING ARTIFICIAL INTELLIGENCE

BY DR. EMRE TUNCEL
EDITED BY ROBERT ZALISK

EDITOR'S NOTE: *Previous AIR Currents have described AIR's portfolio optimization solutions for insurers and reinsurers, and for managing wind pool risk.[1] In this article, Emre Tuncel, Risk Consultant, Consulting and Client Services, explains how catastrophe risk management can be improved using AI approaches to portfolio optimization.*

Artificial Intelligence (AI) techniques can be used to design and build intelligent "agents" that can accomplish specific tasks efficiently. AIR is pioneering the application of AI techniques to solve portfolio optimization problems for the insurance industry using a branch of AI known as *Reinforcement Learning* (RL). RL methodologies are commonly used in the field of robotics, but they are also being adapted and applied to address large-scale and complex optimization problems.

SEQUENTIAL DECISION-MAKING

Central to achieving a satisfactory solution to a practical problem is deciding how to *frame* the problem. The portfolio optimization problem is frequently formulated as a "0-1 knapsack problem," which is a type of "NP-hard problem" (**N**on-deterministic **P**olynomial-time-hard problem, the most complex problem category in computational complexity theory).

Common risk metrics such as"tail value at risk" (TVaR) and "average annual loss" (AAL) are used by insurance companies to measure the marginal impact of adding a policy into a portfolio. While marginal impact is a good proxy for the short-term implications of deciding to write a policy, it does not reveal long-term implications, such as the adverse effect of stacking a portfolio with highly correlated policies.

AI techniques, however, can take both the short- and long-term implications of decisions into account and can also bring some measure of automation to the policy selection process. To use the AI techniques discussed here, portfolio optimization needs to be formulated as a *sequential decision-making problem*—or, still more specifically, as a *Markov Decision Process* (MDP). An MDP is a mathematical framework for modeling decision-making processes

THE ARTICLE: Describes how portfolio optimization can be framed as a sequential decision-making problem in order to apply AI approaches known as "Reinforcement Learning."

HIGHLIGHTS: The iterative nature of the AI techniques employed in the optimization process allows repeated decisions to converge toward a resolution that achieves maximum rewards. The AI approach, compared in a case study with other optimization methods—"Genetic Algorithms" and "Stochastic Steepest Ascent"—is shown to provide superior results.

and problems in which outcomes are partly stochastic and partly under the decision-maker's control. MDPs are framed as: 4-tuple $[S,A,P(\cdot,\cdot),R(\cdot,\cdot)]$

where:

4-tuple indicates that the problem is an ordered repeating set consisting of 4 elements;

S denotes a finite set of states;

A represents a finite set of actions available in a given state;

P indicates probabilities with respect to S and A such that:

the expression $P(s,s')=P(s_{t+1}=s'|s_t=s,a_t=a)$ is the probability that action a in state s at time t will lead to state s' at time $t+1$;

and R—expressed as above: $R(s,s')$—is the expected immediate reward (reinforcement) received after executing action a in state s and transitioning into state s'.

These relationships are illustrated schematically in Figure 1.

AIR

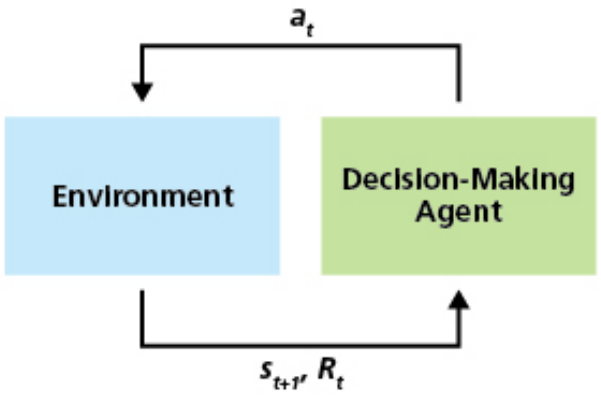

Figure 1. Basic Markov Decision Process (MDP) (Source: AIR)

The basic Markov Decision Process framework is simple: a decision-making agent acts on its environment, receives feedback on whether the action had a positive or negative effect, and selects and executes successive actions one after another (a_t) until a pre-determined stopping condition is met.

Information about the environment is automatically communicated to the decision-making agent (DMA) with each new *t + 1* iteration of an executed action (S_{t+1}). Based on the new state of the environment, the decision-making agent executes an action that causes the environment to transition into a new state in keeping with the transition probabilities, *P(s,s')*. Following this action, the decision-making agent receives a reward or reinforcement (R_t), that reflects the desirability of the new state.

Insurance companies examine common risk metrics such "tail value at risk" (TVaR) and "average annual loss" (AAL) to measure the marginal impact of adding a policy into a portfolio. While marginal impact is a good proxy for the short-term implications of deciding to write a policy, it does not reveal long-term implications, such as the adverse effect of stacking a portfolio with highly correlated policies. AI techniques, however, can take both the short- and long-term implications of decisions into account and can also bring some measure of autonomy to the policy selection process.

ARTIFICIAL INTELLIGENCE FRAMEWORK

The Reinforcement Learning framework makes possible the use of automated optimal decision-making capabilities in uncertain, dynamic environments—such as changing insurance companies' risk profiles. The particular RL algorithm that was used to address the portfolio optimization problem is known as the "Q-Learning algorithm."[2] Q-Learning, like many other RL algorithms, stems from dynamic programming, which makes use of *Bellman Optimality Equations*, which take the form:

$$Q^*(s,a) = E\{r_{t+1} + \gamma \max_{a'} Q^*(s_{t+1},a') \,|\, s_t = s, a_t = a\} \quad (1)$$

$$= \sum_{s'} P^a_{ss'} [R^a_{ss'} + \gamma \max_{a'} Q^*(s',a')] \quad (2)$$

The utility of the Bellman equations is their ability to achieve state-action optimality. In these equations, γ is the discount factor for future rewards and *Q* (s,a)* is the value of the optimal action a that maximizes (or minimizes) the expected immediate reward in state s. In effect, however, RL algorithms adapt the Bellman Optimality Equations into a kind of update rule for the iterative improvement of desired value functions. This "update rule" for basic Q-Learning is a derivative of Equation 2 above and is expressed as follows:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1},a) - Q(s_t,a_t)] \quad (3)$$

Figure 2 adapts the basic Markov Decision Process schematic in Figure 1 to illustrate how the Q-Learning framework is applied to the insurance portfolio optimization problem.

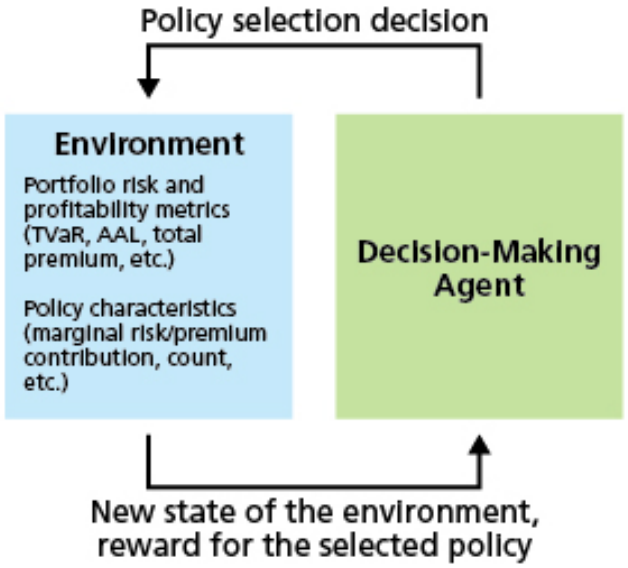

Figure 2. Interaction of the decision-making agent (Markov Decision Process) with an insurance portfolio optimization environment (Source: AIR)

In Figure 2, the "environment" is a portfolio or set of portfolios and the "decisionm-making agent" is a Q-Learning application. The MDP executes a policy selection decision—which immediately changes the state of the (portfolio/policy) environment. That new "state" is conveyed back to the MDP—along with the "reward" information as to whether the change advances toward (or away from) the optimization goal.

ACCOMMODATING UNCERTAINTY

Both the occurrence and frequency of catastrophic events are uncertain, as are the intensity of the events and the damage and loss caused by them. A solution methodology that is able to account for this uncertainty—without having to make unrealistic simplifying assumptions—will give decision makers a competitive advantage.

The Q-Learning technique outlined above is neither too sophisticated for a non-specialist user to understand and implement nor unduly limited in its ability to address challenging complex optimization problems. And, the Q-Learning framework has the ability to handle uncertainty, which is probably its most important advantage.

Q-Learning is considered to be one of the most important breakthroughs in Reinforcement Learning. Q-Learning operates, in effect, by "looking" one step ahead (or more, depending on the problem). In this way, the value of an action *a* in state *s* at time *t* converges toward the value of the action that consistently yields the maximum reward in the new state, $s_{t+1}=s'$, that the environment transitions into at time $t + 1$.

This iterative process is illustrated in Figure 3.

The decision-making agent (Markov Decision Process) in Figure 3 receives portfolio performance information (Step 1) and, based on that information, selects the "best fit" policy from a pool of available policies (Step 2). That selection changes the state of the portfolio, and that new state—along with the valuation that it is desirable or not—is conveyed back to the decision-making agent (Step 3). Finally, valuation and new state information is stored and, a new selection is made (Step 4).

Importantly, the logic of this algorithm allows Q-Learning to operate without having to employ the transition probability and immediate reward information that have been used in dynamic programming. These probability and reward factors have often been criticized as being unrealistic, thus rendering whatever solution methodologies the algorithm arrives at as impractical.

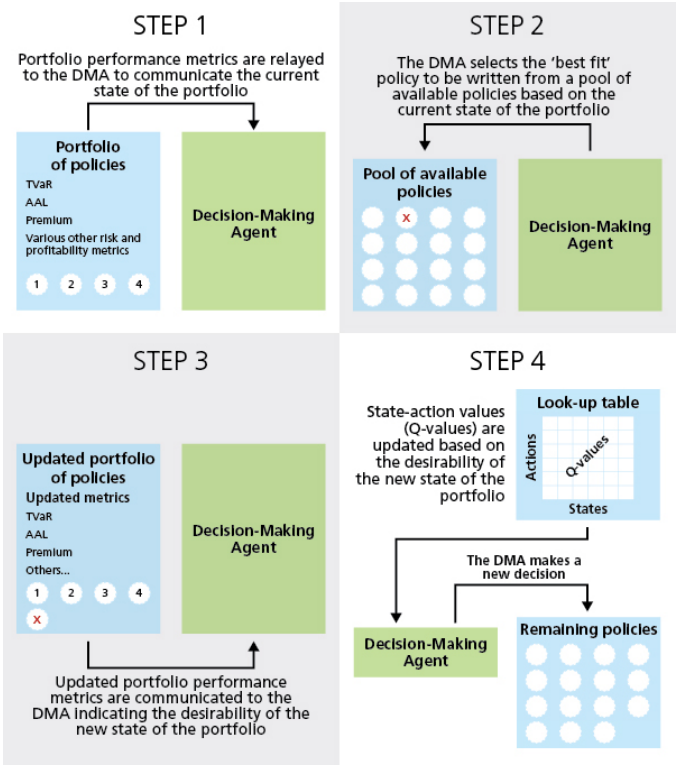

Figure 3. Idealized Q-Learning iterative cycle in execution (Source: AIR)

A CASE STUDY

An experimental case study was undertaken to compare the performance of the Q-Learning portfolio-optimization method with that of other heuristic algorithms, namely Genetic Algorithms (GA) and Stochastic Steepest Ascent (SSA). The data for this case study consisted of 500 policy groups from a residential book in Florida. The premise was that an insurer wanted to identify policy groups suitable for incorporation into its overall portfolio. The insurer also wanted to be able to minimize its exposure while achieving, at the least, a threshold level of premium.

The total premium and TVaR for the entire book were USD 26,074,040 and USD 104,164,319, respectively. This simply means that the TVaR risk metric ranges from 0—if the insurer does not write anything—to USD 104,164,319—if the insurer writes all of the available policy groups—while the amount of premiums that can be collected ranges from 0 to USD 26,074,040. To produce a profit, the insurer needs to collect at least USD 8,500,000 in premiums.

This situation describes an optimization problem for which the goal is to minimize the objective function of TVaR while satisfying the constraint: $Premium_{TOTAL} \geq$ USD 8,500,000.

Because of the stochastic nature of all three optimization methods, the performance comparison between them was conducted as a single-factor two-level experiment that was run 50 times. The mean and standard deviation of the TVaR produced by each method were determined and a confidence interval was computed for the difference in the TVaRs between Q-Learning and GA, the two best-performing approaches, to quantify how much the expectations differ.

The statistical parameters of the final TVaR values for all three methodologies are listed in Table 1, while Figure 4 presents the minimum TVaR yielded by each method after each run.

Table 1. Statistical parameters of the final TVaR values

	MEAN (USD)	STD. DEVIATION (USD)
Q-LEARNING FRAMEWORK	7,423,433	19,645
GA	7,913,001	26,701
SSA	8,269,521	43,060

The confidence interval for the paired difference between the TVaRs yielded by the Q-Learning and the GA is 98%. In other words, 98% of the time the Q-Learning yielded TVaR values between USD 567,934 and 411,203 less than those produced by the GA.

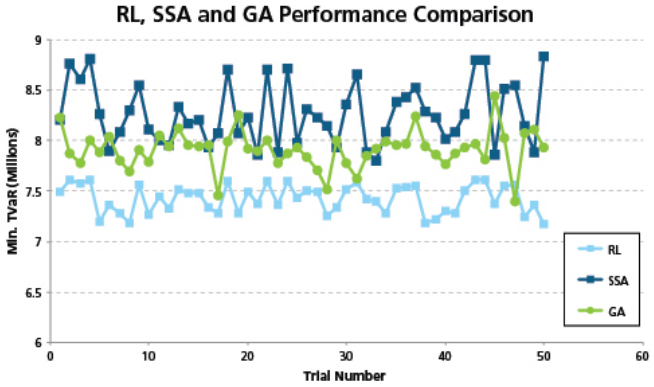

Figure 4. Benchmarking performance (Source: AIR)

The significance of this difference is that the Q-Learning approach yields a better result.

CONCLUSION

Reinforcement Learning techniques, by virtue of their ability to adapt to a stochastic environment, have the potential to advance the insurance portfolio optimization task by delivering superior solutions in the face of uncertainty. Tested against the commonly used Genetics Algorithm in optimizing a book of policy groups, Q-Learning was found to deliver statistically significant superior TVaRs while achieving similar premium levels.

[1]See "Portfolio Optimization for Insurance Companies," January 2011; "Portfolio Optimization for Reinsurers," March 2012; "Managing Wind Pool Risk with Portfolio Optimization," June 2012.

[2]For more information on Q-Learning, see "Reinforcement Learning: An Introduction" by Richard S. Sutton and Andrew G. Barto.

ABOUT AIR WORLDWIDE

AIR Worldwide (AIR) is the scientific leader and most respected provider of risk modeling software and consulting services. AIR founded the catastrophe modeling industry in 1987 and today models the risk from natural catastrophes and terrorism in more than 90 countries. More than 400 insurance, reinsurance, financial, corporate, and government clients rely on AIR software and services for catastrophe risk management, insurance-linked securities, detailed site-specific wind and seismic engineering analyses, agricultural risk management, and property replacement-cost valuation. AIR is a member of the Verisk Insurance Solutions group at Verisk Analytics (Nasdaq:VRSK) and is headquartered in Boston with additional offices in North America, Europe, and Asia. For more information, please visit **www.air-worldwide.com**.