An Adaptive Pursuit Strategy for Allocating Operator Probabilities

Dirk Thierens

Department of Computer Science
Universiteit Utrecht, The Netherlands
Adaptive Operator Allocation

Probability Matching

Adaptive Pursuit Strategy

Experiments

Conclusion

Outline

Adaptive Operator Allocation

Probability Matching

Adaptive Pursuit Strategy

Experiments

Conclusion

Outline

Adaptive Operator Allocation

Probability Matching

Adaptive Pursuit Strategy

Experiments

Conclusion

Dirk Thierens

Adaptive Pursuit Strategy for Allocating Operator Probabilities

Adaptive Operator Allocation: What?

- **Given:**
  - Set of $K$ operators $A = \{a_1, \ldots, a_K\}$
  - Probability vector $P(t) = \{P_1(t), \ldots, P_K(t)\}$: probability of applying operator $a_i$ at time $t$ in proportion to probability $P_i(t)$
  - Environment returns rewards $R_i(t) \geq 0$

- **Goal:** Adapt $P(t)$ such that the expected value of the cumulative reward $E[R] = \sum_{t=1}^{T} R_i(t)$ is maximized

Adaptive Operator Allocation: Why?

- Probability of applying an operator
  - difficult to determine a priori
  - depends on current state of the search process

→ Adaptive allocation rule specifies how probabilities are adapted according to the performance of the operators
Adaptive Operator Allocation: Requirements

- Non-stationary environment ⇒ operator probabilities need to be adapted continuously
- Stationary environment ⇒ operator probabilities should converge to best performing operator

→ conflicting goals!

Probability Matching: Main Idea

- Adaptive allocation rules found in GA literature belong to the class of probability matching strategies
- Main idea: update $P(t)$ such that the probability of applying operator $a$ matches the proportion of the estimated reward $Q_i(t)$ to the sum of all reward estimates $\sum_{i=1}^{K} Q_i(t)$

Probability Matching: Reward Estimate

- The adaptive allocation rule computes an estimate of the rewards received when applying an operator
- In non-stationary environments older rewards should get less influence
- Exponential, recency-weighted average ($0 < \alpha < 1$):

$$Q_a(t + 1) = Q_a(t) + \alpha[R_a(t) - Q_a(t)]$$

Probability Matching: Probability Adaptation

- In non-stationary environments the probability of applying any operator should never be less than some minimal threshold $P_{\text{min}} > 0$
- For $K$ operators maximal probability $P_{\text{max}} = 1 - (K - 1)P_{\text{min}}$
- Updating rule for $P(t)$:

$$P_a(t + 1) = P_{\text{min}} + (1 - K \cdot P_{\text{min}}) \frac{Q_a(t)}{\sum_{i=1}^{K} Q_i(t)}$$
Adaptive Operator Allocation
Probability Matching
Adaptive Pursuit Strategy
Experiments
Conclusion

Main idea
Reward estimate
Probability adaptation
Algorithm
Problem

Adaptive Operator Allocation
Probability Matching
Adaptive Pursuit Strategy
Experiments
Conclusion

Main idea
Reward estimate
Probability adaptation
Algorithm
Problem

Adaptive Operator Allocation
Probability Matching
Adaptive Pursuit Strategy
Experiments
Conclusion

Main idea
Reward estimate
Probability adaptation
Algorithm
Problem

Adaptive Operator Allocation
Probability Matching
Adaptive Pursuit Strategy
Experiments
Conclusion

Main idea
Reward estimate
Probability adaptation
Algorithm
Problem

Adaptive Pursuit Strategy: Pursuit Method

- The pursuit algorithm is a rapidly converging algorithm for learning automata proposed by Thathachar and Sastry
- Main idea: update \( P(t) \) such that the operator \( a^* \) that currently has the maximal estimated reward \( Q_{a^*}(t) \) is pursued
- To achieve this, the pursuit method increases the selection probability \( P_{a^*}(t) \) and decreases all other probabilities \( P_a(t), \forall a \neq a^* \)
- Here, the pursuit algorithm is extended to make it applicable in non-stationary environments

Adaptive Pursuit Strategy: Adaptive Pursuit Method

- Similar to probability matching:
  - The adaptive pursuit algorithm proportionally selects an operator to execute according to the probability vector \( P(t) \)
  - The estimated reward of the selected operator is updated with:
    \[
    Q_a(t + 1) = Q_a(t) + \alpha[R_a(t) - Q_a(t)]
    \]
- Different from probability matching:
  - Selection probability vector \( P(t) \) is adapted in a greedy way

Probability Matching: Algorithm

\[
\text{PROBABILITYMATCHING}(P, Q, K, P_{\text{min}}, \alpha)
\]
1. for \( i \leftarrow 1 \) to \( K \)
2. do \( P_i(0) \leftarrow \frac{1}{K}, Q_i(0) \leftarrow 1.0 \)
3. while NOTTERMINATED()
4. do \( a^s \leftarrow \text{PROPORTIONALSELECTOPERATOR}(P) \)
5. \( R_{a^s}(t) \leftarrow \text{GETREWARD}(a^s) \)
6. \( Q_{a^s}(t + 1) = Q_{a^s}(t) + \alpha[R_{a^s}(t) - Q_{a^s}(t)] \)
7. for \( a \leftarrow 1 \) to \( K \)
8. do \( P_a(t + 1) = P_{\text{min}} + (1 - K \cdot P_{\text{min}}) \frac{Q_a(t)}{\sum_{i=1}^{K} Q_i(t)} \)

- Assume one operator is consistently better
- For instance, 2 operators \( a_1 \) and \( a_2 \) with constant rewards \( R_1 = 10 \) and \( R_2 = 9 \)
- If \( P_{\text{min}} = 0.1 \) we would like to apply operator \( a_1 \) with probability \( P_1 = 0.9 \) and operator \( a_2 \) with \( P_2 = 0.1 \).
- Yet, the probability matching allocation rule will converge to \( P_1 = 0.52 \) and \( P_2 = 0.48 \)!
Adaptive Pursuit Strategy: Probability Adaptation

- The selection probability of the current best operator \( a^* = \arg\max_a [Q_a(t+1)] \) is increased \((0 < \beta < 1)\):
  \[
P_{a^*}(t+1) = P_{a^*}(t) + \beta [P_{\text{max}} - P_{a^*}(t)]
\]

- The selection probability of the other operators is decreased:
  \[
  \forall a \neq a^* : P_a(t+1) = P_a(t) + \beta [P_{\text{min}} - P_a(t)]
  \]

Adaptive Pursuit Strategy: Algorithm

\[
\begin{align*}
\text{ADAPTIVEPURITYT}(P, Q, K, P_{\text{min}}, \alpha, \beta) \\
1 & P_{\text{max}} \leftarrow 1 - (K-1)P_{\text{min}} \\
2 & \text{for } i \leftarrow 1 \text{ to } K \\
3 & \text{do } P_i(0) \leftarrow \frac{1}{K}; Q_i(0) \leftarrow 1.0 \\
4 & \text{while } \text{NOTTERMINATED}() \\
5 & \text{do } a^* \leftarrow \text{PROPORTIONALSELECTOPERATOR}(P) \\
6 & \quad R_{a^*}(t) \leftarrow \text{GETREWARD}(a^*) \\
7 & \quad Q_{a^*}(t+1) = Q_{a^*}(t) + \alpha [R_{a^*}(t) - Q_{a^*}(t)] \\
8 & \quad a^* \leftarrow \text{ARGMAX}_a(Q_a(t+1)) \\
9 & \quad P_{a^*}(t+1) = P_{a^*}(t) + \beta [P_{\text{max}} - P_{a^*}(t)] \\
10 & \text{for } a \leftarrow 1 \text{ to } K \\
11 & \text{do if } a \neq a^* \\
12 & \quad \text{then } P_a(t+1) = P_a(t) + \beta [P_{\text{min}} - P_a(t)]
\end{align*}
\]

Adaptive Pursuit Strategy: Example

- Consider again the 2-operator stationary environment with \( R_1 = 10 \) and \( R_2 = 9 \) \((P_{\text{min}} = 0.1)\).
  As opposed to the probability matching rule, the adaptive pursuit method will play the better operator \( a_1 \) with maximum probability \( P_{\text{max}} = 0.9 \).
  It also keeps playing the poorer operator \( a_2 \) with minimal probability \( P_{\text{min}} = 0.1 \) in order to maintain its ability to adapt to any change in the reward distribution.
We consider an environment with 5 operators $a_i : i = 1 \ldots 5$. Each operator $a_i$ receives a uniformly distributed reward $R_i$ between the boundaries $R_i = U[i - 1 \ldots i + 1]$.

<table>
<thead>
<tr>
<th>Operator reward</th>
<th>$[0..1]$</th>
<th>$[1..2]$</th>
<th>$[2..3]$</th>
<th>$[3..4]$</th>
<th>$[4..5]$</th>
<th>$[5..6]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$R_2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$R_3$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$R_4$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$R_5$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

After a fixed time interval $\Delta T$ the reward distributions are randomly reassigned to the operators.

Upper bounds to performance

If we had full knowledge of the reward distributions and their switching pattern we could always pick the optimal operator $a^\star$ and achieve an expected reward $\mathbb{E}[R_{Opt}] = 5$.

The performance in the stationary (non-switching) environment of a correctly converged operator allocation scheme represents an upper bound to the optimal performance in the switching environment.

3 allocation strategies:
- Non-adaptive, equal-probability allocation rule
- Probability matching allocation rule ($P_{min} = 0.1$)
- Adaptive pursuit allocation rule ($P_{min} = 0.1$)

Non-adaptive, equal-probability allocation rule

The probability of choosing the optimal operator $a^\star_{Fixed}$:

$$\text{Prob}[a^s = a^\star_{Fixed}] = \frac{1}{K} = 0.2$$

The expected reward:

$$\mathbb{E}[R_{Fixed}] = \sum_{a=1}^{K} \mathbb{E}[R_a] \text{Prob}[a^s = a] = \sum_{a=1}^{K} \frac{\mathbb{E}[R_a]}{K} = 3$$

Probability matching allocation rule

The probability of choosing the optimal operator $a^\star_{ProbMatch}$:

$$\text{Prob}[a^s = a^\star_{ProbMatch}] = P_{min} + \left(1 - K \cdot P_{min}\right) \frac{\mathbb{E}[R_{Opt}]}{\sum_{a=1}^{K} \mathbb{E}[R_a]} = 0.2666 \ldots$$

The expected reward:

$$\mathbb{E}[R_{ProbMatch}] = \sum_{a=1}^{K} \mathbb{E}[R_a] \text{Prob}[a^s = a] = \sum_{a=1}^{K} a\left[P_{min} + \left(1 - K \cdot P_{min}\right) \frac{\mathbb{E}[R_a]}{\sum_{a=1}^{K} \mathbb{E}[R_a]}\right] = 3.333 \ldots$$
Adaptive pursuit allocation rule

The probability of choosing the optimal operator $a^*_\text{AdaPursuit}$:

$$\text{Prob}[a^s = a^*_\text{AdaPursuit}] = 1 - (K - 1) \cdot P_{\text{min}}$$

The expected reward:

$$E[R_{\text{AdaPursuit}}] = \sum_{a=1}^{K} E[R_a] \cdot \text{Prob}[a^s = a] = P_{\text{max}} E[R_{a^*}] + P_{\text{min}} \sum_{a=1, a \neq a^*}^{K} E[R_a] = 0.6$$
Class of operator allocation method traditionally used: probability matching
⇒ low probability of applying best operator and low expected reward

Here, adaptive pursuit method introduced
⇒ higher probability of applying best operator and higher expected reward, while ability to react swiftly at environmental changes remains intact

Future work