Design a Cross-Layer Cognitive Engine using Cross-Layer Optimization with Case-Based Reasoning and Reinforcement Learning

4th Workshop of COST Action IC0902
Rome, Italy

09 – 11, October, 2013

Ali Haider Mahdi, Andreas Mitschele-Thiel
Ilmenau, Germany
Motivation

- Efficiently usage of spectrum
- Avoiding interfering with PUs
- Fast CR’s link adaptation according to channel behavior
- Ensure QoS at CR
- Autonomous reconfiguration
- Speed up action process
  - Fast convergence
  - Repeat previous actions
  - Re-evaluate actions
Outline

• State of the Art
• Proposed approach:
  – ADPSO
  – CBR
  – Q-Learning
• Simulation scenario
• Evaluation
• Conclusion
## State of the Art

<table>
<thead>
<tr>
<th>Previous Work</th>
<th>Adv.</th>
<th>Disadv.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- Using Learning from previous action</td>
<td>- Physical layer objectives</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Limited learning process</td>
</tr>
<tr>
<td>Learning and inference system</td>
<td>- Inference and learning based on BN</td>
<td>- No Cross-Layer capability</td>
</tr>
<tr>
<td>[2]</td>
<td></td>
<td>- Single Objective (Bit Error Rate)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Single input parameter (SNR)</td>
</tr>
<tr>
<td>Access network [3]</td>
<td>- Cross layer capability</td>
<td>- No learning from previous actions</td>
</tr>
<tr>
<td></td>
<td>- L2 – L7 application</td>
<td>- FLC has difficult knowledge acquisition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- No learning process</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Physical layer objectives</td>
</tr>
</tbody>
</table>
Proposed approach

- Combination: ADPSO, CBR, Q-L
  - ADPSO: proposes link configuration
  - CBR: selects previous link configuration
  - Q-L: Update previous link configuration’s score
Proposed approach: ADPSO


Proposed approach: ADPSO

• Adaptive Discrete Particle Swarm Optimization (ADPSO):
  – Divide fitness space into four regions
  – Modify the velocity coefficients \((c_1, c_2, w)\) according to fitness value

<table>
<thead>
<tr>
<th>Regions</th>
<th>Fitness</th>
<th>(C_1)</th>
<th>(C_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jump-out</td>
<td>0 – 0.2</td>
<td>- 0.1</td>
<td>+ 0.1</td>
</tr>
<tr>
<td>Exploration</td>
<td>0.2 - 0.4</td>
<td>+ 0.1</td>
<td>- 0.1</td>
</tr>
<tr>
<td>Exploitation</td>
<td>0.4 – 0.6</td>
<td>+0.05</td>
<td>- 0.05</td>
</tr>
<tr>
<td>Convergence</td>
<td>0.6 – 1.0</td>
<td>- 0.05</td>
<td>+ 0.05</td>
</tr>
</tbody>
</table>

\[
W(fitness) = \frac{1}{(1 + 1.5 \times e^{-2.6 \times \text{fitness}})}
\]

– Implements Elitist Learning Strategy (ELS)
Proposed approach: CBR

- Case-Based Reasoning (CBR):
  - Implement past experience
  - Speed up convergence
  - Reduce computation efforts

Proposed approach: Q-Learning

- Q-Learning (Q-L):
  - Study the history of channels
  - Learn appropriate action
Proposed approach: Q-Learning

• Q-L:
  – Select similar previous (state-decision) pairs

<table>
<thead>
<tr>
<th>Ch</th>
<th>Noise</th>
<th>Loss</th>
<th>Pt</th>
<th>M</th>
<th>L</th>
<th>fitness</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>-100</td>
<td>90</td>
<td>25</td>
<td>QPSK</td>
<td>600</td>
<td>0.75</td>
</tr>
<tr>
<td>2</td>
<td>-110</td>
<td>85</td>
<td>20</td>
<td>8PSK</td>
<td>700</td>
<td>0.78</td>
</tr>
<tr>
<td>1</td>
<td>-95</td>
<td>88</td>
<td>30</td>
<td>8PSK</td>
<td>850</td>
<td>0.82</td>
</tr>
</tbody>
</table>
Proposed approach: Q-Learning

- Q-L:
  - Select similar previous (state-decision) pairs
  - Evaluate current fitness at Tx and Rx

<table>
<thead>
<tr>
<th>Ch</th>
<th>Noise</th>
<th>Loss</th>
<th>Pt</th>
<th>M</th>
<th>L</th>
<th>fitness</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>-100</td>
<td>90</td>
<td>25</td>
<td>QPSK</td>
<td>600</td>
<td>0.75</td>
</tr>
<tr>
<td>2</td>
<td>-110</td>
<td>85</td>
<td>20</td>
<td>8PSK</td>
<td>700</td>
<td>0.78</td>
</tr>
<tr>
<td>1</td>
<td>-95</td>
<td>88</td>
<td>30</td>
<td>8PSK</td>
<td>850</td>
<td>0.82</td>
</tr>
</tbody>
</table>
Proposed approach: Q-Learning

- Q-L:
  - Select similar previous (state-decision) pairs
  - Evaluate current fitness at Tx and Rx
  - Update total fitness + Rewards

<table>
<thead>
<tr>
<th>Ch</th>
<th>Noise</th>
<th>Loss</th>
<th>Pt</th>
<th>M</th>
<th>L</th>
<th>fitness</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>-100</td>
<td>90</td>
<td>25</td>
<td>QPSK</td>
<td>600</td>
<td>0.75</td>
</tr>
<tr>
<td>2</td>
<td>-110</td>
<td>85</td>
<td>20</td>
<td>8PSK</td>
<td>700</td>
<td>0.78</td>
</tr>
<tr>
<td>1</td>
<td>-95</td>
<td>88</td>
<td>30</td>
<td>8PSK</td>
<td>850</td>
<td>0.82</td>
</tr>
</tbody>
</table>
Proposed approach: Q-Learning

Q-L:
- Select similar previous (state-decision) pairs
- Evaluate current fitness at Tx and Rx
- Update total fitness
- Select best decision (higher fitness)

<table>
<thead>
<tr>
<th>Ch</th>
<th>Noise</th>
<th>Loss</th>
<th>Pt</th>
<th>M</th>
<th>L</th>
<th>fitness</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>-110</td>
<td>85</td>
<td>20</td>
<td>8PSK</td>
<td>700</td>
<td>0.78</td>
</tr>
</tbody>
</table>
## Simulation Scenario

### Simulation parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of channels</td>
<td>5</td>
</tr>
<tr>
<td>Channel bandwidth</td>
<td>100 kHz</td>
</tr>
<tr>
<td>CCC</td>
<td>1</td>
</tr>
<tr>
<td>PU arrival</td>
<td>0.1 - 1.5 ms</td>
</tr>
<tr>
<td>Noise (dBm)</td>
<td>-85 to -100</td>
</tr>
<tr>
<td>Path loss (dB)</td>
<td>80 to 90</td>
</tr>
<tr>
<td>Min. Data Rate (kbps)</td>
<td>100</td>
</tr>
<tr>
<td>Max. Bit Error Rate</td>
<td>$10^{-4}$</td>
</tr>
<tr>
<td>Transmit power (dBm)</td>
<td>0 - 25</td>
</tr>
<tr>
<td>Modulation scheme</td>
<td>PSK</td>
</tr>
<tr>
<td>Modulation index</td>
<td>1, 2, 3, 4</td>
</tr>
<tr>
<td>Packet length (Byte)</td>
<td>100 - 1000</td>
</tr>
</tbody>
</table>
Simulation scenario (cont.)

Metrics:

- Total fitness

\[ f_1 \] - maximum achievable throughput
\[ f_2 \] - minimum achievable delay
\[ f_3 \] - channel availability
\[ f_4 \] - packet loss probability

\[ w_1 = 0.7, \ w_2 = 0.1, \ w_3 = 0.1, \ w_4 = 0.1 \]

\[ f_{total} = w_1 f_1 + w_2 f_2 + w_3 f_3 + w_4 f_4 \]

[0,1]
- Signaling overhead
- Throughput
- Channel usage
Evaluation: Throughput

Throughput Comparison

Throughput (Packets/Sec)

PU Arrival Rate

- adpso
- adpso+cbr
- adpso+cbr+learning
Evaluation: Signaling overhead

![Graph showing Overhead Comparison]

- **overhead(Adpso)**
- **overhead(Adpso + CBR)**
- **overhead((Adpso + CBR + Learning))**

**PU Arrival Rate**

**Overhead in ms**
Evaluation: Channel Usage
Conclusion

• Efficient algorithm for dynamic environment
• Fast autonomous link adaptation
• Low signaling overhead
• Higher throughput
• Best channel selection
References


Thank you for your attention
Questions?? Comments!!!
Optimization (cont.)

- Evaluation:

[Graphs showing optimization results: Maximum $R_b$ and Minimum BER]

Optimization (cont.)

- Evaluation:

Reasoning (cont.)

- **Evaluation:**

![Bar chart showing comparison of transmission time for different methods.]

Evaluation: Fitness value

![Fitness Value Comparison](image)

- **Fitness Value**
- **Handovers**

Legend:
- Adpso
- Adpso with CBR
- Adpso with CBR and Learning
Evaluation: Data Rate

Data Rate Comparison with Same Environment

- Adpso
- Adpso with CBR
- Adpso with CBR and Learning

Noise: -85 dbm
PU Duration: 0.7 Sec

Noise: -90 dbm
PU Duration: 0.6 Sec

Noise: -96 dbm
PU Duration: 0.4 Sec

Noise: -80 dbm
PU Duration: 0.8 Sec

Data Rate [kbps]

0 10 20 30 40 50

Ali Haider Mahdi
Integrated Communication Systems Group
www.tu-ilmenau.de/ics
State of the Art

- Genetic Algorithm (GA) for link adaptation \[1\]
  - Only for static environment
  - No PU activities model

- Energy-efficient packet size optimization\[2\]
  - Fixed packet size – redundant retransmissions
  - Only for static environment

- Adaptive Discrete Particle Swarm Optimization (ADPSO)
  - Algorithm run for every environmental change
  - New configuration for every packet

→ Need to new approach

---


Evaluation 1: Link config.
Evaluation1: Link configuration

- Noise (dBm)
- Loss (dB)
- Data Channel
- Average Transmit Power (dBm)
- Packet Length (Byte)
Time plan

April 2013 – September 2013

- Q-L in CE
- Simulate CE in OMNeT++
- Poster at Sophia Antipolis
- Publications: 1 accepted 2 planned to submit
- Practical implementation (in progress)

October 2013 – May 2014

- Practical implementation
- Writing dissertation
Scenarios

• Objectives [1,2]
  
  ▪ High Throughput  
  \[ f_{Thr_{max}} = \frac{L}{L+O} \frac{(1-BER)^L+O \cdot R_b}{L_{max}^{1-BER} L_{max}+O R_{b_{max}}} \]

  ▪ Low Transmission delay  
  \[ f_{delay_{min}} = \left( \frac{1}{R_{b_{max}}} \frac{L_{min}}{1 + \frac{1}{R_{b} L}} \right) \]
Similarity

- **Input:**
  - Noise \(\rightarrow\) Normalization \(\rightarrow\) \(\text{Norm}(N)\)
  - Loss \(\rightarrow\) Normalization \(\rightarrow\) \(\text{Norm}(L)\)

- **Previous states:**
  - Noise\(_{-t}\) \(\rightarrow\) Normalization \(\rightarrow\) \(\text{Norm}(N_{-t})\)
  - Loss\(_{-t}\) \(\rightarrow\) Normalization \(\rightarrow\) \(\text{Norm}(L_{-t})\)

- **Euclidean Distance (ED)**
  \[
  = \sqrt{\text{Norm}(N) - \text{Norm}(N_{-t}))^2 + \text{Norm}(L) - \text{Norm}(L_{-t})^2}
  \]

- **Similarity**
  \[
  = 1 - ED
  \]
PU activities model

• Generate exponential random variables \cite{1}
  – IDLE $N_i$
  – BUSY $N_b$

• Mean of the exponential random variables
  – $n_i = \text{mean}(N_i)$
  – $n_b = \text{mean}(N_b)$

• Probability of PU state:
  – Probability of IDLE
    • $P_{r\text{idle}} = \frac{n_i}{n_i+n_b}$
  – Probability of BUSY
    • $P_{r\text{busy}} = \frac{n_b}{n_i+n_b}$

\cite{1} M. Oto and O. Akan, “Energy-efficient packet size optimization for cognitive radio sensor networks,” *IEEE Transactions on Wireless Communications*, vol. 11, no. 4, pp. 1544–553, April 2012.
Proposed approach (cont.)

- Rx sends to Tx over CCC
  - ACK/ NACK according to PER
  - Rx sends current environmental factors (Noise, Loss)
  - Current free channels

<table>
<thead>
<tr>
<th>Frame control</th>
<th>Duration ID</th>
<th>RA</th>
<th>ACK/N ACK</th>
<th>Noise</th>
<th>Loss</th>
<th>Free channels</th>
<th>FCS</th>
</tr>
</thead>
</table>

CCC: Common Control Channel
Proposed approach

Cognitive Engine

ADPSO

CBR

Common Control Channel

Achieve the requirements

Configure CR

Cognitive Radio (Tx)

Data channel

Cognitive Radio (Rx)
Evaluation 2: Signaling overhead

- GA based
- ADPSO based
- ADPSO+CBR based
Conclusion

• Efficient approach in time and signaling overhead for dynamic environment

• Works under different scenarios

• Achievements:
  - Dynamic link configuration
  - High convergence values (Fitness values)
  - Low time
  - Low signaling overhead
Simulation

\[ f_{total} = w_1 f_{Thr} + w_2 f_{power} + w_3 f_{BER} + w_4 f_{delay} \]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmit power (dBm)</td>
<td>0 - 25</td>
</tr>
<tr>
<td>Modulation scheme</td>
<td>PSK</td>
</tr>
<tr>
<td>Modulation index</td>
<td>0, 1, 2, 3, 4</td>
</tr>
<tr>
<td>Packet length (Byte)</td>
<td>100 - 1000</td>
</tr>
<tr>
<td>Code rate</td>
<td>1/2 - 7/8</td>
</tr>
<tr>
<td>Channel type</td>
<td>AWGN</td>
</tr>
<tr>
<td>No. of Free channels</td>
<td>16</td>
</tr>
<tr>
<td>Bandwidth of channel</td>
<td>50 kHz</td>
</tr>
<tr>
<td>Noise (dBm)</td>
<td>-85</td>
</tr>
<tr>
<td>Path loss (dB)</td>
<td>90</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scenario</th>
<th>R_b(kbps)</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voice</td>
<td>64</td>
<td>10^{-3}</td>
</tr>
<tr>
<td>Video</td>
<td>500</td>
<td>10^{-4}</td>
</tr>
<tr>
<td>Data</td>
<td>300</td>
<td>10^{-6}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Weights</th>
<th>M_1</th>
<th>M_2</th>
<th>M_3</th>
</tr>
</thead>
<tbody>
<tr>
<td>w_1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.8</td>
</tr>
<tr>
<td>w_2</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>w_3</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>w_4</td>
<td>0.7</td>
<td>0.7</td>
<td>0</td>
</tr>
</tbody>
</table>
Weights and QoS requirements

<table>
<thead>
<tr>
<th>Weights</th>
<th>Voice</th>
<th>Video</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_1$</td>
<td>0.1</td>
<td>0.1</td>
<td>0.8</td>
</tr>
<tr>
<td>$w_2$</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>$w_3$</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>$w_4$</td>
<td>0.7</td>
<td>0.7</td>
<td>0</td>
</tr>
</tbody>
</table>

Link’s objectives:
- High throughput ($f_1$)
- Low transmission delay ($f_2$)
- Low BER ($f_3$)
- Low power consumption ($f_4$)

Total fitness:
- $f_{total} = \sum_{i=1}^{4} w_i f_i$

<table>
<thead>
<tr>
<th>Scenario</th>
<th>$R_b$(kbps)</th>
<th>BER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voice</td>
<td>64</td>
<td>$10^{-3}$</td>
</tr>
<tr>
<td>Video</td>
<td>500</td>
<td>$10^{-4}$</td>
</tr>
<tr>
<td>Data</td>
<td>300</td>
<td>$10^{-6}$</td>
</tr>
</tbody>
</table>
Results: Convergence in voice scenario

Weights and GA model are in [2]

<table>
<thead>
<tr>
<th>Weights</th>
<th>( W_1 )</th>
<th>( W_2 )</th>
<th>( W_3 )</th>
<th>( W_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voice</td>
<td>0.1</td>
<td>0.7</td>
<td>0.1</td>
<td>0.1</td>
</tr>
</tbody>
</table>