Towards Designing Optimal Individualized Placement Tests

Octavio Lerma\textsuperscript{1}, Olga Kosheleva\textsuperscript{2}, Shahnaz Shahbazova\textsuperscript{3}, and Vladik Kreinovich\textsuperscript{1}

\begin{itemize}
\item \textsuperscript{1}Computational Science Program
\item \textsuperscript{2}Department of Teacher Education
\item University of Texas at El Paso
\item El Paso, TX 79968, USA
\item lolerma@episd.org, olgak@utep.edu, vladik@utep.edu
\item \textsuperscript{3}Azerbaijan Technical University
\item Baku, Azerbaijan, shahbazova@gmail.com
\end{itemize}
1. Need for a Placement Test

- Computers enable us to provide individualized learning, at a pace tailored to each student.
- In order to start the learning process, it is important to find out the current level of the student’s knowledge.
- Usually, such placement tests use a sequence of $N$ problems of increasing complexity.
- If a student is able to solve a problem, the system generates a more complex one.
- If a student cannot solve a problem, the system generates an easier one, etc.
- Once we find the exact level of student’s knowledge, the actual learning starts.
- It is desirable to get to actual leaning as soon as possible, i.e., to minimize the number of placement problems.
2. Bisection – Optimal Search Procedure

- At each stage, we have:
  - the largest level $i$ at which a student can solve, &
  - the smallest level $j$ at which s/he cannot.
- Initially, $i = 0$ (trivial), $j = N + 1$ (very tough).
- If $j = i + 1$, we found the student’s level of knowledge.
- If $j > i + 1$, give a problem on level $m \overset{\text{def}}{=} (i + j)/2$:
  - if the student solved it, increase $i$ to $m$;
  - else decrease $j$ to $m$.
- In both cases, the interval $[i, j]$ is decreased by half.
- In $s$ steps, we decrease the interval $[0, N + 1]$ to width $(N + 1) \cdot 2^{-s}$.
- In $s = \lfloor \log_2(N + 1) \rfloor$ steps, we get the interval of width $\leq 1$, so the problem is solved.
3. Need to Account for Discouragement

- Every time a student is unable to solve a problem, he/she gets discouraged.
- In bisection, a student whose level is 0 will get $\approx \log_2(N + 1)$ negative feedbacks.
- For positive answers, the student simply gets tired.
- For negative answers, the student also gets stressed and frustrated.
- If we count an effect of a positive answer as one, then the effect of a negative answer is $w > 1$.
- The value $w$ can be individually determined.
- We need a testing scheme that minimizes the worst-case overall effect.
4. Analysis of the Problem

• We have $x = N + 1$ possible levels of knowledge.

• Let $e(x)$ denote the smallest possible effect needed to find out the student’s knowledge level.

• We ask a student to solve a problem of some level $n$.

• If s/he solved it (effect = 1), we have $x - n$ possible levels $n, \ldots, N$.

• The effect of finding this level is $e(x - n)$, so overall effect is $1 + e(x - n)$.

• If s/he didn’t (effect $w$), his/her level is between 0 and $n$, so we need effect $e(n)$, with overall effect $w + e(n)$.

• Overall worst-case effect is $\max(1 + e(x - n), w + e(n))$.

• In the optimal test, we select $n$ for which this effect is the smallest, so $e(x) = \min_{1 \leq n < x} \max(1 + e(x - n), w + e(n))$. 
5. Resulting Algorithm

• For $x = 1$, i.e., for $N = 0$, we have $e(1) = 0$.

• We know that $e(x) = \min_{1 \leq n < x} \max(1 + e(x - n), w + e(n))$.

• We can use this formula to sequentially compute the values $e(2), e(3), \ldots, e(N + 1)$.

• We also compute the corresponding minimizing values $n(2), n(3), \ldots, n(N + 1)$.

• Initially, $i = 0$ and $j = N + 1$.

• At each iteration, we ask to solve a problem at level $m = i + n(j - i)$:
  
  – if the student succeeds, we replace $i$ with $m$;
  
  – else we replace $j$ with $m$.

• We stop when $j = i + 1$; this means that the student’s level is $i$. 
6. Example 1: $N = 3, w = 3$

- Here, $e(1) = 0$.
- When $x = 2$, the only possible value for $n$ is $n = 1$, so
  
  $$e(2) = \min_{1 \leq n < 2} \{ \max\{1 + e(2 - n), 3 + e(n)\} \} =$$
  
  $$\max\{1 + e(1), 3 + e(1)\} = \max\{1, 3\} = 3.$$

- Here, $e(2) = 3$, and $n(2) = 1$.
- To find $e(3)$, we must compare two different values $n = 1$ and $n = 2$:
  
  $$e(3) = \min_{1 \leq n < 3} \{ \max\{1 + e(3 - n), 3 + e(n)\} \} =$$
  
  $$\min\{\max\{1 + e(2), 3 + e(1)\}, \max\{1 + e(1), 3 + e(2)\}\} =$$
  
  $$\min\{\max\{4, 3\}, \max\{1, 6\}\} = \min\{4, 6\} = 4.$$

- Here, min is attained when $n = 1$, so $n(3) = 1$. 
7. Example 1: \( N = 3, w = 3 \) (cont-d)

- To find \( e(4) \), we must consider three possible values \( n = 1, n = 2, \) and \( n = 3 \), so

\[
e(4) = \min_{1 \leq n < 4} \{ \max\{1 + e(4 - n), 3 + e(n)\}\} =
\]

\[
\min\{\max\{1 + e(3), 3 + e(1)\}, \max\{1 + e(2), 3 + e(2)\}, \max\{1 + e(1), 3 + e(3)\}\} =
\]

\[
\min\{\max\{5, 3\}, \max\{4, 6\}, \max\{1, 7\}\} =
\]

\[
\min\{5, 6, 7\} = 5.
\]

- Here, \( \min \) is attained when \( n = 1 \), so \( n(4) = 1 \).
8. Example 1: Resulting Procedure

• First, \( i = 0 \) and \( j = 4 \), so we ask a student to solve a problem at level \( i + n(j - i) = 0 + n(4) = 1 \).

• If the student fails level 1, his/her level is 0.

• If s/he succeeds at level 1, we set \( i = 1 \), and we assign a problem of level \( 1 + n(3) = 2 \).

• If the student fails level 2, his/her level is 1.

• If s/he succeeds at level 2, we set \( i = 2 \), and we assign a problem of level \( 2 + n(3) = 3 \).

• If the student fails level 3, his/her level is 2.

• If s/he succeeds at level 3, his/her level is 3.

• We can see that this is the most cautious scheme, when each student has at most one negative experience.
9. **Example 2: \( N = 3 \) and \( w = 1.5 \)

- We take \( e(1) = 0 \).
- When \( x = 2 \), then

\[
e(2) = \min_{1 \leq n < 2} \{ \max\{1 + e(2 - n), 3 + e(n)\} \} =
\]

\[
\max\{1 + e(1), 1.5 + e(1)\} = \max\{1, 1.5\} = 1.5.
\]

- Here, \( e(2) = 1.5 \), and \( n(2) = 1 \).
- To find \( e(3) \), we must compare two different values \( n = 1 \) and \( n = 2 \):

\[
e(3) = \min_{1 \leq n < 3} \{ \max\{1 + e(3 - n), 1.5 + e(n)\} \} =
\]

\[
\min\{\max\{1+e(2), 1.5+e(1)\}, \max\{1+e(1), 1.5+e(2)\}\} = \\
\min\{\max\{2.5, 1.5\}, \max\{1, 3\}\} = \min\{2.5, 3\} = 2.5.
\]

- Here, \( \min \) is attained when \( n = 1 \), so \( n(3) = 1 \).

\[\]

10. Example 2: \( N = 3 \) and \( w = 1.5 \) (cont-d)

- To find \( e(4) \), we must consider three possible values \( n = 1, n = 2, \) and \( n = 3 \), so

\[
e(4) = \min_{1 \leq n < 4} \{ \max\{1 + e(4 - n), 1.5 + e(n)\}\} = \\
\min\{\max\{1 + e(3), 1.5 + e(1)\}, \max\{1 + e(2), 1.5 + e(2)\}, \\
\max\{1 + e(1), 1.5 + e(3)\}\} = \\
\min\{\max\{3.5, 1.5\}, \max\{2.5, 3\}, \max\{1, 4\}\} = \\
\min\{3.5, 3, 4\} = 3.
\]

- Here, \( \min \) is attained when \( n = 2 \), so \( n(4) = 2 \).
11. Example 2: Resulting Procedure

- First, $i = 0$ and $j = 4$, so we ask a student to solve a problem at level $i + n(j - i) = 0 + n(4) = 2$.

- If the student fails level 2, we set $j = 2$, and we assign a problem of level $0 + n(2) = 1$:
  - if the student fails level 1, his/her level is 0;
  - if s/he succeeds at level 1, his/her level is 1.

- If s/he succeeds at level 2, we set $i = 2$, and we assign a problem at level $2 + n(2) = 3$:
  - if the student fails level 3, his/her level is 2;
  - if s/he succeeds at level 3, his/her level is 3.

- We can see that in this case, the optimal testing scheme is bisection.
12. A Faster Algorithm May Be Needed

• For each $n$ from 1 to $N$, we need to compare $n$ different values.

• So, the total number of computational steps is proportional to $1 + 2 + \ldots + N = O(N^2)$.

• When $N$ is large, $N^2$ may be too large.

• In some applications, the computation of the optimal testing scheme may takes too long.

• For this case, we have developed a faster algorithm for producing a testing scheme.

• The disadvantage of this algorithm is that it is only asymptotically optimal.
13. A Faster Algorithm for Generating an Asymptotically Optimal Testing Scheme

- First, we find the real number $\alpha \in [0, 1]$ for which $\alpha + \alpha^w = 1$.
- This value $\alpha$ can be obtained, e.g., by applying bisection to the equation $\alpha + \alpha^w = 1$.
- At each iteration, once we know bounds $i$ and $j$, we ask the student to solve a problem at the level

$$m = \lfloor \alpha \cdot i + (1 - \alpha) \cdot j \rfloor.$$

- This algorithm is similar to bisection, expect that bisection corresponds to $\alpha = 0.5$.
- This makes sense, since for $w = 1$, the equation for $\alpha$ takes the form $2\alpha = 1$, hence $\alpha = 0.5$.
- For $w = 2$, the solution to the equation $\alpha + \alpha^2 = 1$ is the well-known golden ratio $\alpha = \frac{\sqrt{5} - 1}{2} \approx 0.618$. 
14. Acknowledgments

This work was supported in part by the National Science Foundation grants:

- HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and

- DUE-0926721.