Orthogonal Array Testing

Orthogonal Array Testing (OAT) is software testing technique that uses orthogonal arrays to create test cases. It is statistical testing approach especially useful when system to be tested has huge data inputs. Orthogonal array testing helps to maximize test coverage by pairing and combining the inputs and testing the system with comparatively less number of test cases for time saving.

For example, when a train ticket has to be verified, factors such as - the number of passengers, ticket number, seat numbers, and train numbers have to be tested. One by one testing of each factor/input is cumbersome. It is more efficient when the QA engineer combines more inputs together and does testing. In such cases, we can use the Orthogonal Array testing method.

This type of pairing or combining of inputs and testing the system to save time is called Pairwise testing. OATS technique is used for pairwise testing.

Why OAT (Orthogonal Array Testing)?

In the present scenario, delivering a quality software product to the customer has become challenging due to the complexity of the code.

In the conventional method, test suites include test cases that have been derived from all combination of input values and pre-conditions. As a result, n number of test cases has to be covered.

But in a real scenario, the testers won't have the leisure to execute all the test cases to uncover the defects as there are other processes such as documentation, suggestions, and feedback from the customer that has to be taken into account while in the testing phase.

Hence, the test managers wanted to optimize the number and quality of the test cases to ensure maximum Test coverage with minimum effort. This effort is called Test Case Optimization.

1. Systematic and Statistical way to test pairwise interactions
2. Interactions and Integration points are a major source of defects.
3. Execute a well-defined, concise of test cases that are likely to uncover most (not all) bugs.
4. Orthogonal approach guarantees the pairwise coverage of all variables.

How OAT's is represented

The formula to calculate OAT

- Runs (N) – Number of rows in the array, which translates into a number of test cases that will be generated.
- Factors (K) – Number of columns in the array, which translates into a maximum number of variables that can be handled.
- Levels (V) – Maximum number of values that can be taken on any single factor.

A single factor has 2 to 3 inputs to be tested. That maximum number of inputs decide the Levels.

How to do Orthogonal Array Testing: Examples

1. Identify the independent variable for the scenario.
2. Find the smallest array with the number of runs.
3. Map the factors to the array.
4. Choose the values for any "leftover" levels.
5. Transcribe the Runs into test cases, adding any particularly suspicious combinations that aren't generated.

Example 1

A Web page has three distinct sections (Top, Middle, Bottom) that can be individually shown or hidden from a user

- No of Factors = 3 (Top, Middle, Bottom)
- No of Levels (Visibility) = 2 (Hidden or Shown)
- Array Type = L4(23)

(4 is the number of runs arrived after creating the OAT array)

If we go for Conventional testing technique, we need test cases like 2 X 3 = 6 Test Cases

Test Cases	Scenarios	Values to be tested
Test #1	HIDDEN	Top
Test #2	SHOWN	Top
Test #3	HIDDEN	Bottom
Test #4	SHOWN	Bottom
Test #5	HIDDEN	Middle
Test #6	SHOWN	Middle

If we go for OAT Testing we need 4 Test cases as shown below:

Test Cases	TOP	Middle	Bottom
Test #1	Hidden	Hidden	Hidden
Test #2	Hidden	Visible	Visible
Test #3	Visible	Hidden	Visible
Test #4	Visible	Visible	Hidden

Example 2:

A microprocessor's functionality has to be tested:

1. Temperature: 100C, 150C and 200C.
2. Pressure : 2 psi,5psi and 8psi
3. Doping Amount :4%,6% and 8%
4. Deposition Rate : 0.1mg/s , 0.2 mg/s and 0.3mg/s

By using the Conventional method we need = 81 test cases to cover all the inputs. Let's work with the OATS method:

No. of factors = 4 (temperature, pressure, doping amount and Deposition rate)

Levels = 3 levels per factor (temperature has 3 levels-100C, 150C, and 200C and likewise other factors too have levels)

Create an array as below:

1. Columns with the No. of factors

Test case #	Temperature	Pressure	Doping amount	Deposition rate

2. Enter the number of rows is equal to levels per factor. i.e temperature has 3 levels. Hence, insert 3 rows for each level for temperature,

Test case #	Temperature	Pressure	Doping amount	Deposition rate
1	100C			
2	100C			
3	100C			
4	150C			
5	150C			
6	150C			
7	200C			
8	200C			
9	200C			

3. Now split up the pressure, doping amount and the deposition rates in the columns.

For eg: Enter 2 psi across temperatures 100C,150C and 200C likewise enter doping amount 4% for 100C,150C and 200C and so on.

Test case #	Temperature	Pressure	Doping amount	Deposition rate
1	100C	2 psi	4%	0.1 mg/s
2	100C	5 psi	6%	0.2 mg/s
3	100C	8 psi	8%	0.3 mg/s
4	150C	2 psi	4%	0.1 mg/s
5	150C	5 psi	6%	0.2 mg/s
6	150C	8 psi	8%	0.3 mg/s
7	200C	2 psi	4%	0.1 mg/s
8	200C	5 psi	6%	0.2 mg/s
9	200C	8 psi	8%	0.3 mg/s

Hence, in OAs, we need 9 Test cases to cover.

OAT Advantages

- Guarantees testing of the pair-wise combinations of all the selected variables.
- Reduces the number of test cases

- Creates fewer Test cases which cover the testing of all the combination of all variables.
- A complex combination of the variables can be done.
- Is simpler to generate and less error-prone than test sets created by hand.
- It is useful for Integration Testing.
- It improves productivity due to reduced test cycles and testing times.

OAT Disadvantages

- As the data inputs increase, the complexity of the Test case increases. As a result, manual effort and time spent increases. Hence, the testers have to go for Automation Testing.
- Useful for Integration Testing of software components.

Mistakes or errors while performing OAT

1. The testing effort should not be focused on the wrong area of the application.
2. Avoid picking the wrong parameters to combine
3. Avoid using Orthogonal Array Testing for minimal testing efforts.
4. Applying Orthogonal Array Testing manually
5. Applying Orthogonal Array Testing for high-risked applications

Conclusion:

Here we have seen how OAT (Orthogonal Array Testing) can be used to reduce the testing efforts and how test case optimization can be achieved.