Alexander Booth
Found Visualize
October 23, 2017

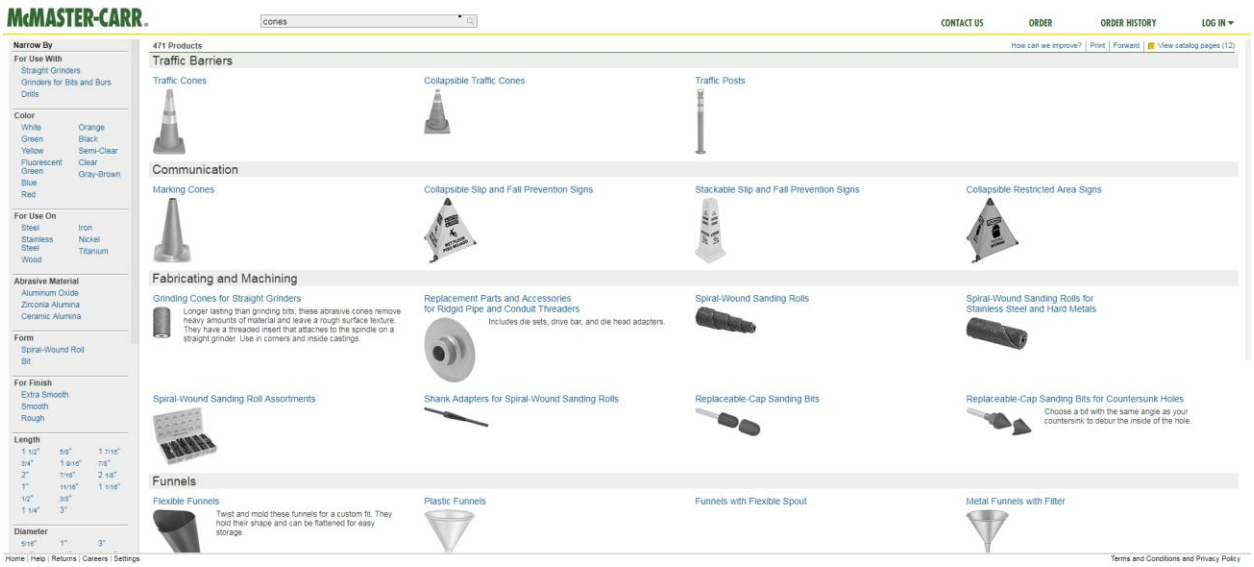
Concurrent A/B Testing

Purpose:
The success of our A/B testing program relies on the number of tests run per year, the percentage of winning tests, and average customer impact per successful experiment. By severely limiting the number of tests we run for the sake of avoiding data pollution, we are also significantly reducing the velocity of our testing.

Currently, our primary goal is to figure out the validity of a single test, and consequently to be confident in the attribution and the impact of that test. Hence, we usually avoid tests that might have overlapping traffic. However, this means that we are not running efficiently all possible tests that might benefit the customer experience, again reducing our velocity and potentially losing money.

Running multiple simultaneous tests will increase the velocity of our testing program. However, this can pollute our data since multiple separate tests could potentially affect each other's outcomes. This is known as a test interaction, and will be referred to quantitatively as an interaction term.

Example:
Before this document gets into actual testing strategies, let us first contemplate a simple scenario. Below is the current landing page for "cones"



Now, one of the important attributes is color. We feel that colored images would show customers our product breadth and result in more success actions. This will be **Test 1,** the Colored Images test**.** The "A" group will get the current black and white images while the "B" group will see colored images.

Further, we develop a hypothesis that removing copy from all groups will create a sleeker page and encourage more clicks and consequently success actions from customers. This will be **Test 2,** the No Copy test. The "C" group will see the landing page with copy, while the "D" group will see no copy.

For both of these tests, we need 1000 customers to see each one, 500 per group, and we know this means the tests will take around 3 weeks.

Caveat:
The "A", "B", "C", and "D" groups mentioned here are distinctions between two A/B tests and should not be confused with one A/B/C/D test.

Strategies:
The first strategy is perhaps the most simple.

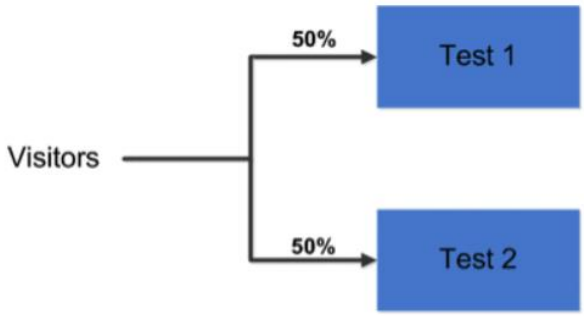
1) Sequential

Perform the two tests sequentially. Since the two tests are on the same page, there will probably some interaction term between the changes. The safest course of action is always to run them sequentially, or one after the other. This isolates the changes and mitigates any influences one test has on the other. However, this will take 6 weeks plus the time it takes to set up and tear down tests as well as implement the winner of Test 1. One drawback of this strategy is that our results could change depending on the order we run the test.

We perform the Color Test, and determine that color images, ("B") are in fact best. Analyzing and implementing "A", as well as setting up the next test takes about a week. Then we run the No Copy Test and determine that the No Copy ("D") betters the customer experience. After a combined 7 weeks, we show a combination of "BD" on the cones page.

2) Mutually Exclusive (You Got to Keep Them Separated)

In this strategy, traffic is split between the tests. This means that all visitors will see either a variant of Test 1 and the default for Test 2, or the default for Test 1 and a variant for Test 2. All misleading effects between variants, the interaction terms, are thus avoided by separating the traffic.



Since it takes 3 weeks for 1000 customers to visit the cones page, and we need 1000 people for each test, this strategy will take 6 weeks. Since we don't have to implement the winner in the middle, this time to completion is shorter than the Sequential strategy. Unfortunately, there is a large drawback.

We decide to implement the Mutually Exclusive strategy for our two tests on the cone page. After 6 weeks, 1000 people have each seen the black and white, or colored images, with Copy, the default of the second test. Another 1000 people have seen Copy or No Copy, with black and white images, the default of the second test. After analyzing the two tests independently, it is decided that Colored images and No Copy are both the winner. However, **nobody has seen this combination.** Only AC, BC, and AD were tested, not BD. This means we have no clue how visitors will behave when they see the variants from both tests, since no visitor traffic had seen that combination during our testing.

A spin-off of this strategy is the aptly named
2.5) Don't Worry, It'll Be Fine. Let's Just Do It. What's the Worst that Could Happen?

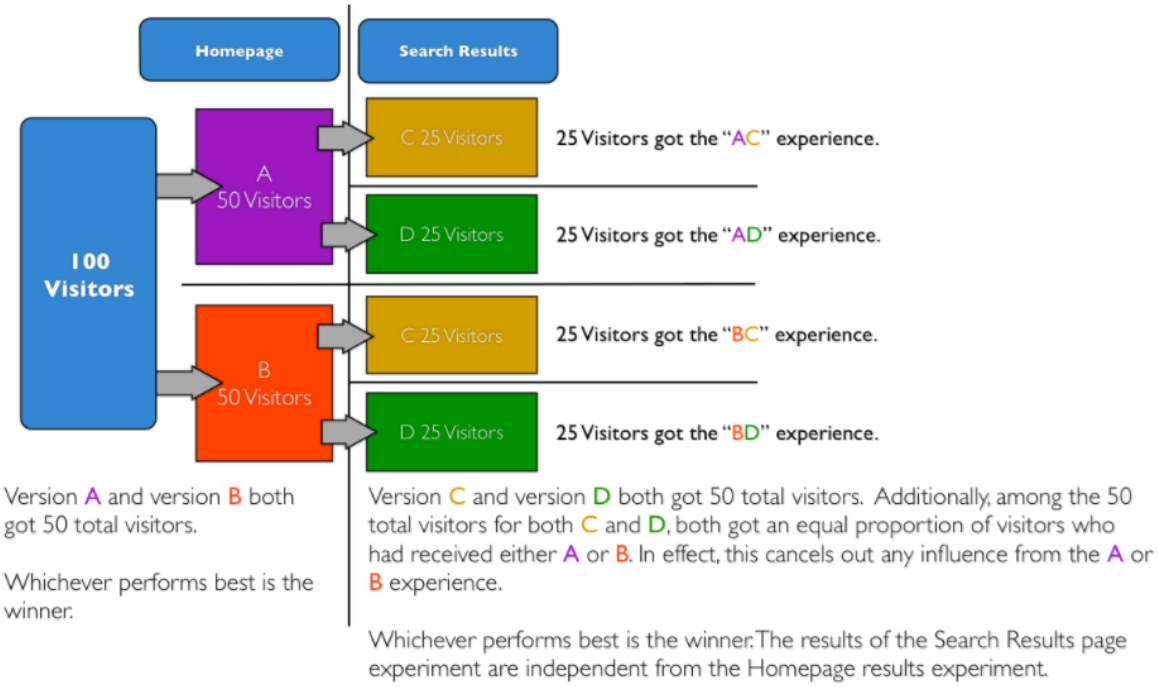
This is the most common, and easiest, strategy. It just assumes that the different tests don't really impact one another, and that we can run each test without bothering to account for the other one. This strategy entails launching two A/B test on two disparate pages with unique populations and no relationship between them. These requirements can also be slightly lenient. In reality, this is often fine in many cases, especially if there is limited overlap.

For example, imagine we want to test black and white vs colored images on the cones page and small vs wide spec search in Activity. We can run both of these at the same, analyze them separately, and develop individual conclusions as they meet the requirements outlined above. Additionally, it will only take as long as one test. Even if both variants are chosen, it shouldn't matter that nobody saw the combination.

To deal with the case of launching an untested combination of changes, this next strategy was developed:

3) Stratified Testing
"Even if one test's variation is having an impact on another test's variations, the effect is proportional on all the variations in the latter test and therefore, the results should not be materially affected." – L. IAR

Since the interaction effect between tests is equal, we can stratify our populations equally across the tests. Half of the customers who see version "A" in the first test will see version "C" in the second test. The other half will see version "D". The same logic applies for version "B". The result is a stratified testing population where both tests still receive the required amounts of visitors and the interaction between tests is mitigated, at least in theory. The results can also be determined in the time of only one test, making it the fastest option by far.



For our example, over three weeks, 1000 customers visit the cones page. 500 of them get version "A" in the Colored Images test. Of those 500, 250 get version "C" of the No Copy test. The other 250 get the "D" version of the No Copy test. The same logic applies to the "B" version of the Colored Images test. At the end of three weeks, 500 people have seen each of "A" and "B", as well as "C" and "D". Since we stratified the populations, we can disregard the interaction term and just choose the best performer for each test, of which some customers would have seen a combination.

Results 1:
Test 1: Colored Image Test

- Version "A" had a 10% success rate
- Version "B" had a 15% success rate

Test 2: No Copy Test

- Version "C" had a 15% success rate
- Version "D" had a 20% success rate

Simply by looking at these numbers, we would release a combination of version "BD" to the cones page.

Does this sound too good to be true? Well that's **because it is**.

The underlying assumption is a dangerous one. That is, the assumption that whatever effect Test 1 might have on Test 2, it has the same effect on all of the variants of Test 2.

So what's wrong with this assumption? Great question, reader.

It is possible that the "interactions" between variants in the two tests are not equal to each other and uniformly spread out. Let me change the results from our stratified example above:

Results 2:
Test 1: Colored Image Test

- Version "A" had a 10% success rate
- Version "B" had a 10% success rate

Test 2: No Copy Test

- Version "C" had a 15% success rate
- Version "D" had a 15% success rate

Looking at both tests individually, the success rates for both tests are equal between the default and the variant. We would thus conclude that the default wins in both cases, since the variant or challenger did not achieve a higher success rate. We would release version "AC" to the cones page.

However, if we look at the visitors for Test 2 who saw the Variant from Test 1, compared with the visitors for Test 2 who saw the default for Test 1, we see that the success rates are different. In fact, we could actually get a **higher success rate** overall if we implement the variants from Test 1 and Test 2.

TEST 1	**SR**	**TEST 2**	**SR**
Default "A"	10%	Default "C"	13%
		Variant "D"	10%
Variant "B"	10%	Default "C"	17%
		Variant "D"	20%

Without looking at the split across tests, version "C" had an average success of 15% and version "D" also had an average success of 15%, giving us our results from Result 2. However, this is obviously a case where the interaction effects between tests matter and this often goes unnoticed. Nevertheless, it can have a major impact on our test conclusions.

Although this strategy gets answers faster than any other option, its accuracy can be wildly off.

All is not lost. There is another option.

4) Multivariate Testing

The final strategy discussed here is to combine the tests and run them as a single multivariate test. Multivariate tests can be setup for tests running on a single page or across multiple pages. With a multivariate test we can test all possible experiences together, which in our example would be the four experiences "AC", "AD", "BC", "BD".

A multivariate test allows us to find out which combination of variants yields the highest success rates as well as how much each individual variant contributes to that rate. In addition, multivariate tests can be

optimized, where experiences are automatically removed and narrowed to a winning combination. This process can reduce the number of experiences tested over time. Furthermore, this strategy is the most accurate way to run the tests, as visitors will see all combinations of variants systematically. However, it does still take a while to run both tests as the number of visitors to each variant needs to be acquired to make the results statistically significant. Further, if we are testing more than two variables, then the combinations, and time to run the test, increase exponentially.

For our example, it would take 6 weeks to conclude the test. We need 500 people to visit each variant. If we have 1000 customers every 3 weeks visiting the cones page, then it would take 6 weeks to reach the required number of visitors. The results end up only differing slightly from Result 2 of the stratified test, due to the increased number of visitors to make each group statistically significant.

Results 3:

Test Combination	**SR**
AC	12%
AD	11%
BC	18%
BD	19%

Based off of these results, we would launch combination “BD” to the cones page.

While this strategy takes the same amount of time as strategy two, Mutually Exclusive, and only slightly shorter than running the tests sequentially, it provides the most accurate results for combinations of variants and individual variant contributions.

Conclusions

When it comes to running tests concurrently we need to weigh the options. Some tests may influence each other more than others. In this document, I have summarized the different options and the points that we should consider. Below, I have summarized them based upon their speed to get answers and the accuracy of the results.

Option	Speed	Accuracy
Run the Tests Sequentially	Low	Medium
Run Tests at the Same Time	High	Low
Run Tests at the Same Time (Splitting Traffic)	Medium	Medium
Combine Tests & Run as MVT	Medium	High

Knowing the options and the risks for each is vitally important.

Further Reading

https://blogs.oracle.com/marketingcloud/running-multiple-tests-simultaneously
https://conductrics.com/ab-testing-when-tests-collide-2/
https://conversionxl.com/blog/can-you-run-multiple-ab-tests-at-the-same-time/
https://www.optimizely.com/optimization-glossary/multivariate-testing/
https://en.wikipedia.org/wiki/Multivariate_testing_in_marketing