

The Truth About Taguchi

Why Fractional Factorial Multivariate Testing is Wrong for Landing Page Optimization

Tim Ash - President, SiteTuners.com

Brought to you by:



The following whitepaper uses extensive quotations from the book *Landing Page Optimization: The Definitive Guide to Testing and Tuning for Conversions* by SiteTuners.com president Tim Ash (John Wiley & Sons Press, 2008). Copyright © 2008 by Wiley Publishing, Inc., Indianapolis, Indiana. All Rights Reserved. Reprinted with permission.



For additional information please visit:

LandingPageOptimizationBook.com

SiteTuners.com

Executive Summary

Landing page optimization and testing is a powerful way to improve the profitability of your online marketing programs. By testing alternative presentations of information in your mission-critical online processes on your audience you can dramatically improve the conversion rate of these desired actions (such as sales, registrations, form-fills, or downloads).

The most basic method of landing page testing is a simple head-to-head test of your original landing page against an alternative version. Beyond such "A-B Split testing" there are a number of more-powerful multivariate testing methods that allow you to consider multiple changes to the landing page at once.

With the growing popularity of landing page optimization, certain approaches have been canonized and have taken on an almost mythical reverence among the ranks of online marketers. It is almost as if the buzzwords themselves confer some special power on the practitioner (e.g., "Design of Experiments", "fractional factorial," and the "Taguchi method").

In reality there is a huge mismatch between the original environment in which such fractional factorial testing was developed and how it is usually applied to landing page optimization. It was basically transplanted to online marketing because it is relatively easy for a nonmathematical audience to understand, and not because of its appropriateness or fitness for the task.

The principal drawbacks of fractional factorial methods are:

- o Very small test sizes
- o Restrictive & inflexible test designs
- o Less accurate estimation of individual variable contributions
- o Drawing the wrong conclusions
- o Inability to consider context and variable interactions

Despite misinformation to the contrary, fractional factorial methods do not offer any data collection speed advantage over similar full factorial data-collection approaches (such as those available in the Google Website Optimizer tool). If you plan on using parametric (i.e. "model building") approaches for landing page testing you should use full factorial data collection regardless of the subsequent analysis you plan to do.

All parametric methods (including fractional factorial) are also outclassed by newer non-parametric testing methods, which have the following advantages:

1. Very large test sizes (1000-10,000 times larger with the same data rate)
2. Much faster data collection (on the same data rate)
3. More accurate results (consider variable interactions)
4. Flexible test construction

Landing page testing consists of two main parts: deciding what to test, and finding the best solution among your available options. Those who still insist that fractional factorial methods are terrific because they have produced significant conversion rate improvements are confused. The improvements in these tests are a direct result of the quality of the ideas tested, and not of the inaccurate fractional factorial methods used to find the best version of the landing page.

1. Basic Terminology

Before I discuss common testing methods, you need to understand some common concepts and definitions used in landing page optimization. Members of the testing community refer to the same concepts but use different language to describe them. I will note such cases as appropriate, and will use the terminology interchangeably for the remainder of this paper.

The primary objective of landing page testing is to predict the behavior of your audience given the specific content on the landing page that they see. You will collect a limited sample of data during your test, summarize and describe it, and predict how people from the same traffic sources will act when interacting with the page. The ultimate goal is to find the best possible version of the landing page among all of the variations that you are testing.

Input and Output Variables

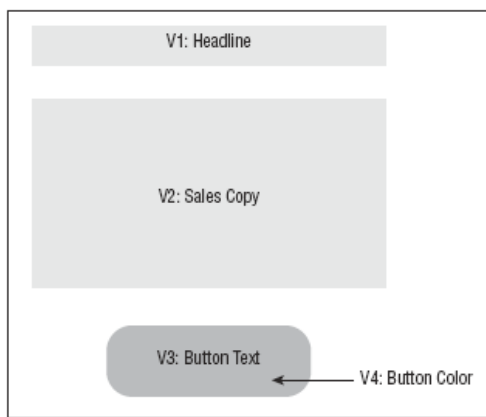
A landing page test has two basic components: a set of *input variables* (also called “independent variables”) that you can control and manipulate, and one or more *output variables* (or “dependent variables”) that you measure and observe. Note that what I mean by “independent variables” here is not the same as in the discussion of variable interactions later in this paper. Independent variables as discussed here are simply the specific page elements that you have chosen for your test.

Variable

The word *variable* (when used by itself) means a page element that you have selected. Variables can be of any granularity or coarseness. For example, a variable might be the headline of your landing page, or a whole-page redesign. In multivariate testing, a variable is also commonly referred to as a *factor*.

In a multivariate test, you will have more than one variable. To distinguish among them I will use the following notation: a capital “V” followed by the number that you have assigned to a particular variable. For example, let’s assume that you have a simple landing page with a headline, some sales copy text, and a call-to-action on a button (see below). You might decide to test alternatives to each of these page elements and name them as follows:

- V1 = Headline
- V2 = Sales copy
- V3 = Button text
- V4 = Button color



Note that the variables do not necessarily define a unique physical location on the page. In fact, V3 (the button text) and V4 (the button color) actually occupy the same space. Nor do they have to be localized. For example, I can choose a variable to test a larger font size (for improved readability) versus an existing smaller one. In this case, the font size change would take effect throughout my whole landing page and would overlap with other variables (such as the actual text on the page) that I might also be testing.

Value

A value is a particular *state* that a variable can take on. When traditional multivariate testing is used in other fields, variable values are often *continuous* (which means they can vary smoothly across a range). This allows you to predict the behavior at interpolated values of the variable (in between the places where you actually sample). For example, if I know that the output of a car engine at 1000 RPM (revolutions-per-minute) is 100 horsepower, and at 2000 RPM is 200 horsepower, I can interpolate between these two values to estimate that the output should be 150 horsepower at 1500 RPM.

In landing page tuning, variable values are almost always *discrete* (distinct from each other, and countable). For example, a button color might be green, blue, or red. I will number the possible choices by successive lowercase letters. By convention, the letter *a* represents the original version of the variable (as seen on your baseline pretest landing page). The letter is combined with the variable name to exactly specify the value of a particular variable. If V4 is our button color, an example assignment might look as follows:

V4a = green button (the original)
V4b = blue button
V4c = red button

Unlike continuous variables, measuring the effect of discrete variable values does not give us any information about the other possible values. Continuing our example from earlier, even if we had measured the average conversion rates with the green and blue buttons, we would not have any information about the performance of the red one.

Branching Factor

The total number of possible values for a discrete variable is called its *branching factor*. For discrete variables, the branching factor must be at least 2 (the original version and one alternative). As I will discuss later in this paper, some testing methods require that the branching factor be the same for all variables in the test. In the button color example, V4 has a branching factor of 3 (because it can take on the values signified by a-green, b-blue, and c-red). In traditional multivariate testing, the number of values for a variable is called the *level of the factor*. Each value is also called a *level* because historically it was drawn from continuous variables. For example, if your variable only has two values, they might be signified by "low" and "high" (or "-1" and "+1").

Recipe

A *recipe* is a unique combination of variable values in your test. It is a sequential listing of the specific values that each variable takes on in the specific version of the landing page.

For example, let's assume that you had set the following variable values from my previous example for a particular landing page in your test: V1b, V2c, V3a, V4a. The recipe could be abbreviated as *bcaa*.

Each recipe is unique. By convention, the recipe with all *a*'s is the original or *baseline* recipe to which all others will be compared.

2. Overview of Multivariate Testing

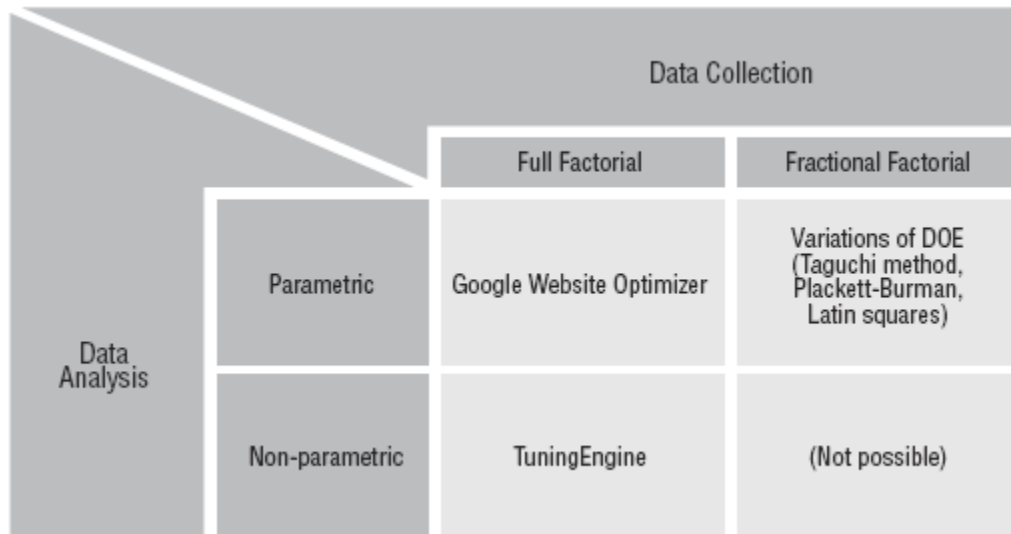
The purpose of *multivariate testing* is to simultaneously gather information about multiple variables, and then conduct an analysis of the data to determine which recipe results in the best performance.

Multivariate testing approaches differ on three important dimensions:

- How the test is constructed (deciding what to test)
- How the data is collected
- How the data is analyzed

The data can be collected in a *full factorial* or *fractional factorial* fashion (see the "Data Collection" section below). The subsequent analysis can be either *parametric* or *non-parametric* (see the "Data Analysis" section below). Within parametric analysis there are also significant differences. Some forms of parametric analysis take complex variable interactions into account, while others do not.

I have presented test construction, data collection and data analysis as independent dimensions in the sections below. In fact, they cannot always be separated. The choice of one can dictate the choice of the others. If you choose an arbitrary test construction, you may not be able to use fractional factorial approaches. Likewise, if you choose fractional factorial data collection and test construction, you automatically lock yourself into a very restricted subset of parametric models for your subsequent data analysis (i.e., non-parametric analysis is impossible if you conduct fractional factorial data collection). The figure below summarizes the possible multivariate testing approaches.



There are two important considerations when picking a testing method:

- The size of your test (expressed as the total number of possible recipes)
- The ability to consider variable interactions

These will be discussed in more detail later in the paper. The table below summarizes available testing methods along these dimensions.

Testing Method	A/B Split	Fractional Factorial Parametric	Full Factorial Parametric	Non-Parametric
Search Space Size (Typical)	1-10 recipes	10-100 recipes	10-100 recipes	1,000,000+ recipes
Considers Variable Interactions	No	No	Yes	Yes

3. Test Construction

When you are deciding how to construct your test, there is an important distinction that must be made: Is your test structured or unstructured?

Unstructured designs are by far the simplest to implement. You simply choose exactly how many variables you want to test, and the branching factor for each one. As long as you are reasonable, you can choose very different numbers of values for each variable. For example, you can pick 7 different headline, 2 button colors, and 9 calls-to-action. This allows you to pay particular attention to the test elements that you think will result in the greatest performance improvements, and devote more of your test sampling to finding the best values for them. The branching factor is simply a function of how many good creative alternatives you want to test for a particular variable. If you choose full factorial data collection, you will enjoy the benefits of unstructured designs.

Structured designs are an artifact of fractional factorial data collection. By assuming a specific underlying model, they force you to have a specific shape to your test. In other words, the number of variables and their branching factors are predefined, and can not be violated. For example, you might be forced to have one variable with a branching factor of two and seven variables with a branching factor of three. There are a number of standard test constructions to choose from, but you must use one of them.

4. Data Collection

Full factorial experimental designs sample data across your whole search space. If this is done properly, the subsequent analysis allows you to consider not only the main effects, but all variable interactions as well (including higher-order ones).

Technically, all *fractional factorial* designs fall under the *design of experiments (DOE)* umbrella. DOE is a systematic approach to getting the maximum amount of useful information about the process that you are studying, while minimizing the amount of effort (measured in unique recipes sampled) and data collection required.

By definition, fractional factorial experimental designs make simplifying assumptions about the possible form of the parametric model for subsequent analysis. For example, they may simply assume that there are no higher-order interactions at all (i.e., that the values of the corresponding coefficients in the model are zero). In the extreme case, they can assume that only the main effects matter and that there are no interactions of any kind (even lower-order

ones). As you will see in the "Variable Interactions" section below, making assumptions about the underlying process that may not be valid and may actually lead you astray.

5. Data Analysis

Parametric data analysis in landing page optimization builds a model of how the variables tested (the "independent variables") impact the conversion rate (the "dependent variable"). For each recipe in your search space, the model will produce a prediction of the expected conversion rate (or other optimization criterion of interest). Unless you happened to have sampled data on the exact recipe predicted by the model as being the best, you do not really know if the prediction will hold up. That is why it is critical to run follow-up A-B split tests between the predicted best challenger recipe and the original baseline recipe for all parametric data analysis methods.

By contrast, *non-parametric* data analysis does not try to build a model based on the input variables. Non-parametric methods try to identify the best challenger recipe, but without being able to tell anything about *why* it is the best, or *exactly how much better* it is than your baseline.

The two approaches are unrelated and are answering different questions. They are both a recognition of the fundamental reality that only so much useful information can be extracted from your data collection sample. The only question is what you want to do with the data. You can try to create a general model of the output variable and try to describe it in terms of the input variables, or you can find the best individual recipe and not know why it is the best.

Parametric Analysis

After you collect your data, you can build a model that expresses how your dependent variable (e.g., the conversion rate) varies based on the settings of your independent variables (i.e., your tuning elements and their specific values). The models are made up of two types of components. *Main effects* describe the impact of an individual variable value on the results. In other words, they look at each variable in isolation, and see how changing its value affects the results. *Interaction effects* consider combinations of variable values, and how they influence each other when presented together. Interaction effects are possible among two or more variable values. For example, if you had five variables in your test, you could have interactions involving any subset of two, three, four, or five variable values. Interactions involving smaller numbers of variables are called lower order, while those involving many variables are called higher order.

As I mentioned earlier, variables are commonly referred to as factors in parametric multivariate testing terminology. Likewise, variable values are often referred to as levels. If a variable has a branching factor of two, the levels are often referred to as "high" and "low" (or are denoted by "+1" and "-1"). Similarly, three levels are often denoted by "+1", "0", and "-1".

Usually parametric multivariate testing uses the general mathematical class of *linear models* based on the analysis of variance (ANOVA). In other words, you are trying to predict the output variable by adding up the contributions of all of the possible main effects and interaction effects of the input variables. You start with the average value of your output variable in the test, and then add in the positive or negative impact of your input variables and their interactions.

Let's consider the simplest possible multivariate example. Assume that you are testing a new call-to-action button and are considering two colors (blue, green) and two font styles (Arial, Times Roman) for the text:

V1a = blue
V1b = green
V2a = Arial
V2b = Times Roman

You can create a model of the conversion rate that "fits" your data as well as possible and uses the average value, main effects, and all interactions. The coefficients (denoted by c 's in front of each effect) indicate the magnitude of the contribution of each effect and can be either positive or negative:

$$CR = c_1 + c_2 \times V1a + c_3 \times V1b + c_4 \times V2a + c_5 \times V2b \\ + c_6 \times V1a:V2a + c_7 \times V1a:V2b + c_8 \times V1b:V2a + c_9 \times V1b:V2b$$

c_1 represents the average value, c_2 – c_5 are the main effects, and c_6 – c_9 are the two variable interaction effects (involving all four possible combinations of the two variables).

Let's assume that your experiment is slightly larger. You now add a third two-way variable to the test (designated by V3a and V3b). The full model with all interactions is shown below (the new terms resulting from the addition of the third variable are bolded).

$$CR = c_1 + c_2 \times V1a + c_3 \times V1b + c_4 \times V2a + c_5 \times V2b \\ + \mathbf{c_6 \times V3a + c_7 \times V3b} \\ + c_8 \times V1a:V2a + c_9 \times V1a:V2b + c_{10} \times V1a:V3a + c_{11} \times V1a:V3b + c_{12} \times V1b:V2a + c_{13} \times V1b:V2b \\ + \mathbf{c_{14} \times V1a:V2a:V3a + c_{15} \times V1a:V2a:V3b + c_{16} \times V1a:V2b:V3a + c_{17} \times V1a:V2b:V3b} \\ + \mathbf{c_{18} \times V1b:V2a:V3a + c_{19} \times V1b:V2a:V3b + c_{20} \times V1b:V2b:V3a + c_{21} \times V1b:V2b:V3b} \\ + \mathbf{c_{22} \times V2a:V3a + c_{23} \times V2a:V3b + c_{24} \times V2b:V3a + c_{25} \times V2b:V3b}$$

As you can see, the number of coefficients that you must now estimate in the model has mushroomed from 9 to 25. For the first time, you see the presence of three variable interaction effects.

The examples above are among the smallest possible multivariate tests. As you can see, if you have a higher branching factor for each variable, or a larger number of variables, the number of coefficient terms in the model grows very quickly. Fractional factorial parametric approaches force you to choose the complexity of your model ahead of time. This means you must somehow decide *in advance* which main effects are important, and also which interactions will be included in the model. In landing page testing this is practically impossible.

Based on the complexity of your parametric model, you can determine its *resolution*. The resolution is a scale that describes your ability to separate out the main effects and lower-order interactions with a particular data collection experimental design. The most common types of designs are resolution III-V. Resolution II designs are not useful because you cannot even estimate the main effects properly. Resolution VI and above are too complex and assume that high-order interactions are common. Higher resolution designs sample across a larger fraction of the whole search space. Simpler resolution III designs are sparse and sample only a small

proportion of the total search space. Most of the common fractional factorial methods for landing page optimization are resolution III designs.

Non-parametric Analysis

It is a practical impossibility to do the following three things simultaneously without a ridiculously large data rate:

1. Find the best performing recipe
2. Search a very large search space
3. Explain why the winning recipe is the best (while ignoring important interactions)

Non-parametric approaches use a different mathematical foundation and starting assumptions to focus on items #1 and 2. They do not assume anything about the underlying model or even the presence or size of variable interactions. Parametric approaches focus on items #1 and 3.

So in effect, parametric versus non-parametric can be viewed as the inherent tradeoff between being able to model and explain the best recipe, and being able to find it in a much larger search space.

Unfortunately, some companies in our field make ludicrous assertions to the contrary. They claim to be able to test a "virtually unlimited number of variations" of a landing page by using a modified fractional factorial approach, while still taking variable interactions into account, and being able to describe which individual values of the winning recipe contributed exactly how much to its improved performance. Ludicrous claims like this make everyone in our industry look bad.

6. Search Space Size

The number of unique recipes in your test is your *search space size*. It can generally be computed by multiplying together all of the branching factors of the variables in your test.

In my earlier example from the "Basic Terminology" section, let's assume that there are three headlines, four versions of the sales copy, four calls-to-action, and three button colors (and "BF" below stands for the branching factor for a particular variable).

$$\begin{aligned}\text{Search space size} &= \text{BFV1} \times \text{BFV2} \times \text{BFV3} \times \text{BFV4} \\ &= 3 \times 4 \times 4 \times 3 \\ &= 144\end{aligned}$$

This example is a small one. As you can see, if you have more variables, and/or higher branching factors for each one, the search space size will grow very rapidly. If the search space size is large, it can quickly exceed the practical limits of common testing methods such as A-B split testing, fractional factorial parametric, and full factorial parametric testing.

Some testing methods can only scale to very small sizes. Fractional factorial approaches are rarely used for test involving more than a few hundred recipes. Non-parametric methods can often involve millions of recipes. As a rough guide, non-parametric approaches can search through 1000 to 10,000 time more recipes than parametric ones. As a practical matter this may mean testing half a dozen new ideas versus a couple of dozen. Online marketers can quickly exceed the practical size of parametric tests and must then choose which of their ideas they

should include. In effect, they are self-censoring and guessing ahead of time which ones are likely to be the winners - thus undermining the whole premise of testing everything on your audience. By being able to run much larger tests, you effectively democratize the process, allowing many people's input to be used for deciding on the test variables and values. This diversity of opinions and ideas is more likely to lead to better results.

7. Variable Interactions

"A player who makes a team great is much more valuable than a great player."

—UCLA Coach John Wooden

When professional basketball players were first allowed to play in the Olympic Games, the United States assembled a "dream team" from the ranks of top NBA superstars. The expectation was that this high-powered assembly of top talent would walk all over their competition. However, the United States lost in the gold-medal match to Yugoslavia.

How could this have happened? Clearly the individual U.S. players were superior to their Yugoslav counterparts. But the Yugoslav squad had trained together and was used to playing by the slightly different rules of Olympic basketball. By contrast, the U.S. team was assembled shortly before the games and had not practiced very much. They had not "jelled" as a team. Similarly, some of the landing page elements that you may be testing may be superstars *individually*. But you should be looking for the *combination* of variables that performs best when presented together.

What is a *variable interaction*? Simply put, it is when the setting for one variable in your test positively or negatively influences the setting of another variable. If they have no effect on each other, they are said to be *independent*. In a *positive interaction*, two (or more) variables create a synergistic effect (yielding results that are greater than the sum of the parts). In a *negative interaction*, two (or more) variables undercut each other and cancel out some of the individual effects.

Let's look at a simple example. Let's assume that you are an auto dealer who sells both Ferraris and Volvos. Your goal is to sell cars and you want to test two different headlines and two different accompanying pictures. So there are a total of four possible versions based on your two variables.

Recipe *aa*

Ferraris are Really Fast



Recipe *ab*

Ferraris are Really Fast



Recipe *ba*

Volvos are Very Safe



Recipe *bb*

Volvos are Very Safe



If you believe that there are no interactions, then you must also believe that there is a “best” headline regardless of the accompanying picture, and that there is a “best” picture regardless of the headline used.

Clearly this is not the case. Each variable depends on the *context* in which it is seen. Recipe *aa* has a strong positive interaction (connecting the speed and power in the picture with the word “Fast” in the headline). Recipe *ab* has a strongly negative interaction (making you think about the consequences of fast driving—“speed kills”). Recipe *ba* has a mildly positive interaction (supporting the notion that you can go fast and still be safe). Recipe *bb* has a positive interaction (playing on the fear of accidents and highlighting Volvo’s longstanding safety record).

So it’s not the picture, and it’s not the headline that determines the performance of the ad - it is their particular *combination*.

In online marketing, we *want* interactions. We want the picture to reinforce the headline, and the sales copy, and the offer, and the call-to-action... Similarly, we want to detect any parts of the landing page that are working at cross-purposes and undercutting the performance of other page elements. Our goal should be to find the best performing *group* of landing page elements.

Some tuning methods (such as A-B split testing and many forms of fractional factorial parametric testing) assume that there *are absolutely no interactions* among your variables (that they are completely independent of each other).

For online marketing this is an absurd assumption. Very strong interaction effects (often involving more than two variables) definitely exist, and in SiteTuners’ experience are pretty common. This should not be a surprise to anyone, since online marketers are intentionally trying to create landing pages that are greater than the sum of their parts. You should be looking for synergies among all of your page elements and trying to eliminate combinations of variable values that undermine your desired outcome.

So while you may be able to get some positive results by ignoring interactions, you will not be getting the *best* results. So where can you look for interactions? In general, there is no way to guarantee that any subset of your testing elements does not interact. However, you should consider elements that are in physical proximity, or that are otherwise confounded with each other. For example, let’s assume that you are testing a form and have chosen to test the call-to-action button color and text. Although these may seem independent, that is not the case. They both combine to create the specific presentation of the call-to-action, and you should test for possible interactions.

Similarly, if you are testing different headlines followed by different sales copy, you should expect interactions. The headline is supposed to draw the visitor into reading further. If there is a disconnect between the headline and the following text, you can expect negative interactions. If they reinforce each other, you should expect positive synergies. So far I have primarily focused on interactions between two test elements. In fact, there are often strong interactions among several variables on a landing page.

You need to step back and take a critical look at the tuning method you are about to use and its implications for your test. Are parametric fractional factorial testing methods better than A-B split testing, assuming that you have a high enough data rate? Sure. But don't let that blind you to one of their glaring defects—most common fractional factorial techniques do not take variable interactions into account.

As I will discuss in the "Full Factorial Parametric Testing" section below, full factorial data collection does allow you to later examine variable interactions. However, you also need to be clear about full factorial *data collection* versus subsequent full factorial *data analysis*. They are independent of each other.

Currently the Google Website Optimizer is the only widely-used parametric multivariate testing tool that allows you to collect data in a full factorial fashion, but the reporting and analysis still only looks at the main effects. You can export the complete data set and conduct a more complicated analysis, but this is not currently supported in the tool itself.

So the dirty little secret is out. If you still choose to ignore variable interactions, you have no one but yourself to blame for suboptimal results. The bottom line is this: if you do not have the minimum data rate to use non-parametric tuning methods like the TuningEngine (which can handle very large test sizes *and* considers variable interactions), then you should at least use full factorial data collection coupled with a proper subsequent analysis that estimates variable interactions. Identifying interactions can be a complicated and unpleasant business. You may have to learn some additional statistics or bring in outside experts to help you to design the test and analyze your results, but this is the only way to consistently get the biggest possible benefits.

Don't Ignore Variable Interactions

- Interactions exist and can be very strong.
- If you ignore them, you will not get the best results.
- Fractional factorial testing methods generally assume that variable interactions do not exist.

8. Fractional Factorial Parametric Testing

As I mentioned earlier in this paper, fractional factorial data collection can not really be divorced from parametric data analysis. So I will refer to the combination of the two simply as "fractional factorial."

In theory, it is possible that every variable that you test has interactions with every specific value of every other variable. In practice, this is usually not the case. During your test, you may discover that many or even most of the elements that you have decided to include do not impact performance at all. They simply do not matter to your audience. It is also common that strong interactions between two variables exist but that higher-order interactions (among three or more variables) are insignificant. In such cases, the behavior of the output variable can be described by looking at the main effects and a few low-order interactions (involving two

variables). This basic idea arises as a consequence of three empirical principles commonly understood in the testing community.

Hierarchical Ordering Principle

Lower-order effects are more likely to be important than higher-order effects.
Effects of the same order are equally likely to be important.

This principle suggests that when resources are scarce (i.e., the data collection rate is low), priority should be given to estimating main effects and lower-order interactions.

Effect Sparsity Principle

The numbers of relatively important effects in a factorial experiment are small.

This is another formulation of the 80/20 rule. Only a few variables combine to produce the biggest effects, and all of the rest will not matter nearly as much.

Effect Heredity Principle

In order for an interaction to be significant, at least one of its parent factors should be significant.

This is another application of common sense. If a variable does not produce any big effects on its own (i.e., it is benign or negligible), it is unlikely to do so when combined with something else. It may be that a big interaction effect is produced by variables that do not show the largest main effects, but at least one of the variables involved in an interaction will usually show some main effect. The whole idea behind fractional factorial design is that you can collect data on a fraction of the recipes needed for an equivalent full factorial design and still maximize the model's predictive value.

Fractional factorial designs are expressed using the notation $k-p$ (k is the common branching factor for all variables in the test; k is the number of variables investigated, and p describes the size of the fraction of the full factorial search space used). In mathematical terms, p is the number of generators (elements in your model that are confounded and cannot be estimated independently of each other). In other words, when you increase p , you are really saying that some of your input variables are not independent and can be explained by some combination of the other input variables or their interactions.

A design with p generators will require a $1/(l_p)$ fraction of the full factorial design search space size. For example, let's assume that you have a 2_{6-2} fractional factorial design. A full factorial 2_6 experimental design would require you to sample all 64 possible recipes. But the simpler fractional design will require sampling only 16 recipes ($1/4$ of the total). Creating a proper fractional factorial design is beyond the scope of this book. The basic steps are as follows:

- Based on the generators (see above) of your design, you can determine the *defining relation*.
- The defining relation specifies the *alias structure*.
- A fractional factorial experiment is created from a full factorial experiment by using the chosen *alias structure*.

One common constraint on fractional factorial tests is that the branching factor is two for all variables. The methods for creating custom test designs outside of this constraint are complex.

Many testers simply copy “standard” designs from statistical texts, and restrict themselves to a choice of variables and branching factors that fit the model.

Taguchi method background and basics

Although there are some difference among these common fractional factorial methods, their basic predictive power, required data sample size, and underlying assumptions are pretty similar. The main difference lies in the test construction and *shape* of the search spaces that each can be used for. So if you are going to use any of these methods, you should base your decision on your familiarity with each and the number and branching factor of the variables in your test. There is no reason to prefer the Taguchi method over Plackett-Burman, or Latin squares, but since it is gaining currency in landing page testing I will focus on it here.

Genichi Taguchi was a Japanese mathematician and proponent of manufacturing quality engineering. He focused on methods to improve the quality of manufactured goods through both statistical process control and specific business management techniques. Taguchi developed many of his key concepts outside of the traditional Design of Experiments (DOE) framework and only learned of it later. His main focus was on robustness—how to develop a system that performed reliably even in the presence of significant noise or variation. In traditional DOE, the goal is to model the best-performing recipe. In other words, the higher the value of the output variable (e.g., the conversion rate), the better. So the goal is to find the highest *mean*. When taking repeated samples, any variation is considered a problem or a nuisance.

Taguchi had a different perspective. He felt that manufacturing quality should be measured by the amount of deviation from the desired value. In other words, he was concerned not only with the mean, but also with the amount of *variation* or “noise” produced by changing the input variables. So optimization from the Taguchi perspective means finding the best settings for the input variables, defined as the ones producing the highest signal-to-noise ratio (the highest mean with the least amount of variation). An important consideration is how to keep the noise in the output low even in the face of noisy inputs.

The numbers of variables (factors) and alternative values for each variable (levels) is arbitrary in landing page optimization tests. You can easily find additional variables to test, or come up with alternative values for each variable. Unfortunately, basic Taguchi arrays exist only for the following experimental designs:

- L4—Three two-level factors
- L8—Seven two-level factors
- L9—Four three-level factors
- L12—Eleven two-level factors
- L16—Fifteen two-level factors
- L16b—Five four-level factors
- L18—One two-level and seven three-level factors
- L25—Six five-level factors
- L27—Thirteen three-level factors
- L32—Thirty-two two-level factors
- L32b—One two-level factor and nine four-level factors
- L36—Eleven two-level factors and twelve three-level factors
- L36b—Three two-level and twelve three-level factors
- L50—One two-level factor and eleven five-level factors
- L54—One two-level factor and twenty-five three-level factors
- L64—Twenty-one four-level factors
- L81—Forty three-level factors

These test design arrays can be combined in various ways to create additional Taguchi-compliant experimental designs, but you will probably need the help of a statistician to implement them.

The Taguchi method uses orthogonal arrays that obtain a lot of information about the main effects with a relatively small number of recipes. However, many of his experimental designs are *saturated* (allowing no way to estimate interaction effects).

The Big Mismatch

These problems are a direct consequence of the Taguchi method's origins in manufacturing. Let's take a look at some of the characteristics of this original environment:

- **Expensive prototypes** The underlying assumption is that creating alternative recipes is difficult, time-consuming, or expensive. When applications involve physical processes, human medical trials, or manufacturing technology, this is indeed the case. So the goal is to minimize the required number of recipes (also called "prototypes" or "experimental treatments") in the test.
- **Small test sizes** A direct consequence of the expensive prototypes is that you need to keep the number of elements that you test to an absolute minimum, and focus only on the most critical variables.
- **No interactions** As another consequence of the expensive prototypes, you can only measure the main effects created by your variables. The small test size and expensive data collection force you to assume very sparse fractional factorial models that cannot accurately estimate even two variable interactions.
- **High yields** In most cases, the process or outcome that you were measuring had a high probability of success (a high conversion rate).
- **Continuous variables** Many of the input variables involved in the tests were continuous (e.g., temperature, concentration of a particular chemical compound). Even though you had to pick specific levels of the variable for the test, you could often interpolate between them to estimate what would happen at non-sampled settings of the variable.

These manufacturing approaches were transplanted to the online marketing arena (and landing page optimization in particular) because of their relative simplicity and familiarity. Unfortunately, the assumptions that accompanied them came along for the ride, even though they are not applicable to the new environment.

Let's take a closer look at the reality of landing page testing:

- **Free prototypes** When you create a test plan, you define exactly which page elements to test, and specify the alternative variable values for each. In most cases, the alternative test elements are easy to implement. They involve changes to the HTML structure of your page, text changes, and graphics.

Once this preliminary work has been done, you have the capability to create any of the recipes in your search space. In other words, the process of creating a different version of your page is completely automated. There is no incremental cost to showing *any* of the possible recipes to your next visitor. So there is no need to restrict yourself to showing only a

small percentage of the possible recipes. Recipes are free to create and display to your landing page visitors.

- **Huge test sizes** If you critically reviewed your landing page, you were probably able to identify dozens of potential problems (large and small) with it. For each of your original test elements, you can probably come up with several alternatives that can reasonably be expected to produce better results and should be included in your test plan. If you did, your search space size would be in the millions (or even billions) of possible recipes. Unfortunately, multivariate testing is specifically designed for very small test sizes. Most real-world tests involve search spaces of a few dozen total recipes.
- **Significant interactions** As discussed earlier, variable interactions play a huge role in landing page optimization. Some interactions are unexpected and are the result of the reduced coherency of the mix-and-match presentation of variables during a test. The effect may be something like Frankenstein's monster—stitched together from functional parts, but not resulting in a very appealing whole.

Other variable interactions are intentional. In fact, as an online marketer you should want to create interactions. You should look for page elements that work together and support each other in getting your visitor to act. These kinds of synergies are at the very heart of good marketing. Yet most fractional factorial designs assume that there are no interactions (or that they are very small).

- **Low yields** Some landing pages have double-digit conversion rates, but most pages have lower rates. Many e-commerce websites have conversion rates that are well below 1%. The limiting factor in the length of the data collection for these pages is the number of conversions (rather than the number of recipes sampled in the test).
- **Discrete variables** Most of the elements that you test on a landing page are discrete. They involve completely distinct choices that are unrelated to each other. For example, if you tested a particular headline for your page, you would not be able to predict the performance of an alternative headline.

By now, you have probably determined my preference for full factorial over fractional factorial data collection for landing page optimization if you are going to use parametric data analysis. There is no efficiency disadvantage to full factorial designs during the data collection stage and significant advantages during the analysis stage.

Most Taguchi method test arrays are resolution III designs—and can only estimate the main effects in the model. In other words, they cannot capture all possible two-variable interactions (or any higher-order interactions). Some of them explicitly assume that there are no interactions. They use this radical assumption to dramatically lower the number of sampled recipes and amount of data required to estimate the main effects. An important additional requirement for all of these approaches is that the data collection is balanced across all possible values of a variable (i.e., you cannot use uneven data sampling, or it may complicate or throw off your use of standard data analysis).

Let's assume that you want to collect data for each of the variable main effects in the examples that follow. You can construct a series of increasingly larger tests and see how few recipes you can get away with.

The simplest case is an A-B split test containing two recipes, *a* and *b*. You need to split your traffic 50/50 across *a* and *b*. So you need two recipes to measure the two values of variable V1. These two recipes represent your entire search space.

Now imagine that you have two variables, each with a branching factor of two. This results in four possible recipes: *aa*, *ab*, *ba*, and *bb*. You choose to sample only from recipes *aa*, and *bb* (still only two recipes as in the previous example). Note that half of the data that you collect will involve V1a (from recipe *aa*), and half will involve V1b (from recipe *bb*). Similarly, half of your data will cover V2a (from recipe *aa*), and half will involve V2b (from recipe *bb*). As you can see, you have collected equal amounts of data on each main effect, and you did it by sampling only half of your total search space (two out of four recipes).

Let's extend our example to three variables, each with a branching factor of two. This results in eight possible recipes: *aaa*, *aab*, *aba*, *abb*, *baa*, *bab*, *bba*, and *bbb*. You choose to sample only from recipes *aaa* and *bbb* (still only two recipes). Note that half of the data that you collect will involve V1a (from recipe *aaa*), and half will involve V1b (from recipe *bbb*). Similarly, half of your data will cover V2a (from recipe *aaa*), and half will involve V2b (from recipe *bbb*). Half of your data will also cover V3a (from recipe *aaa*), and half will cover V3b (from recipe *bbb*). You have again collected equal amounts of data on each main effect, and have done it by sampling only a quarter of your total search space (two out of eight recipes).

Of course you cannot continue to sample just two recipes and still cover all main effects at larger test sizes. But by clever test construction, you can keep the number of unique recipes surprisingly small (especially when considered as a proportion of the total search space). If you think that the previous examples are a bit contrived and artificial, you are right.

Underlying the use of fractional factorial methods is the assumption that creating a test run is difficult or time-consuming—so you need to keep the number of recipes that you sample as low as possible. This may have been true in the manufacturing setting (e.g., retooling an assembly line to test for a change in production quality), but it is not true or necessary in landing page optimization. Internet technology allows you to easily create any recipe of your landing page test. The page is dynamically created on the fly for each new visitor. For practical data collection purposes, it does not matter how many unique recipes you have in your test. For the assumption of expensive recipe construction you pay a heavy price during data analysis. By sampling very limited recipes, you destroy your ability to do a comprehensive analysis and find variable interactions later.

Fractional Factorial Disadvantages

Fractional factorial designs have several disadvantages:

Small test size Search spaces are very small (it is rare to see landing page tests with more than a few hundred recipes). In a typical test, you may only be able to explore a few new ideas, and must arbitrarily decide which ones are good enough to be tested. Most online marketers want to run much larger tests. It is common to come up with dozens of alternative variable values for your test after only a short brainstorming session.

Does not consider variable interactions The most common fractional factorial landing page testing approaches assume a model that is simple, in order to capture important variable interactions. As previously discussed, all of these methods are resolution III designs and can only estimate the main effects of your input variables. This can significantly skew the results and lead you to costly incorrect conclusions.

Piecewise construction errors Another common mistake is to take the winning values from each variable and combine them into a single recipe. This piecewise construction does not necessarily constitute the best-performing recipe.

Let's take a closer look at why this is the case. Assume that you have picked a 90% statistical confidence threshold for each variable in your test. In other words, you are 90% sure that the particular value for that variable is the best-performing one. If you had only one variable in your test, you would be wrong 10% of the time, and this might be acceptable to you.

But the likelihood of error grows quickly as you increase the number of variables in your test. For example, in a two-variable test your chances of finding the best recipe depend on you being correct about the best value for each variable independently of the other. This means that you must multiply together the probabilities of being right for each variable.

In our example, this would mean that your chances of finding the correct recipe are 81% (90% x 90%). So your error rate has increased from 10% for a single variable to 19% for two. By the time you get six variables in your test, you are only 53% certain of having found the best recipe. This is only slightly better than flipping a coin. For this reason, it is critical to run a follow-up head-to-head test between your predicted best answer and your original baseline recipe.

But what do you do if the predicted performance of your challenger recipe does not measure up? If this is due to piecewise construction errors, you can raise your confidence threshold, or lower the number of variables in your test. But the unexpectedly poor performance could also be due to huge interaction effects that you have failed to consider. The only way to find these is to rerun the test with a higher resolution design (preferably a full factorial one).

Highly sensitive to streaky data Multivariate test designs are very sensitive to lucky streaks in your data. This is especially a problem if you are collecting very small data samples. For example, let's assume that you are testing a landing page that has a real underlying conversion rate of 1%. Within the first hundred visitors, it is almost equally likely that you would have zero or two conversions. However, the estimates produced by your models would vary drastically in these two cases. The first model would take the data very literally and conclude that the likelihood of conversion with this landing page is zero (i.e., it will never produce a conversion). The second would conclude that your conversion rate is double its actual value. It is critical to collect a lot of data with multivariate testing models to reduce the problems associated with such possible small-sample distortions.

Requires you to guess at important interactions All fractional factorial models require you to specify exactly what types of main effects and variable interactions are possible in your model. These assumptions must be built in ahead of time in order to simplify the complexity of the model and give you economy in terms of the number of recipes that must be sampled.

In the traditional testing, this might be possible since you know which *physical* processes can have an influence on each other. But in online marketing this is difficult. Unlike physical experiments (e.g., in manufacturing, or pharmaceutical drug trials), landing page optimization is trying to tease out the underlying *psychological* predispositions of people. Since everyone is different, it is impossible to accurately empathize with every member of your audience. You cannot take your own predispositions out of the experiment because you are the one choosing the elements to test. In such a setting, it is impossible to declare which specific interactions matter and which others don't.

Restrictive test construction and design As previously discussed, there must be a certain pattern (in terms of the number of variables and their branching factors) in your test design. So you are

forced to either stick with well-known “standard” designs from statistical textbooks or construct your own (with the help of statisticians).

Throttling is very difficult If you “throttled” your data collection rates (i.e. did not devote equal bandwidth to each recipe in your test), your analysis will be invalid for all common fractional factorial designs.

9. Full Factorial Parametric Testing

A full factorial parametric test collects data on the response of every possible combination of variables (factors) and values (levels). In other words, it collects an equal amount of information about every possible recipe in your search space. As I discussed earlier, in online marketing we *expect* strong variable interactions. In fact, we are doing everything that we can to create positive synergies among our tuning elements.

“We at Google are continually surprised by how common and strong variable interactions are in landing page tests.”

—Mike Myer, statistician, Google

Unlike fractional factorial data collection, full factorial data collection does not lock you into any restrictions during analysis. Full factorial parametric tests do not make assumptions about the underlying model. Analysis of your full factorial data can pinpoint your main effects as well as any interaction effects (lower order or higher order) present in your test.

SiteTuners.com is one of the charter Google Website Optimizer Authorized Consultant companies. The Google Website Optimizer is a free A-B split testing and full factorial parametric multivariate testing tool available to the general public. You do not have to spend any money on AdWords to use the tool. We use it to run all smaller multivariate tests in which the data rate is too low to apply our proprietary non-parametric TuningEngine technology. I am very glad that Google has chosen the full factorial data collection as their default.

There is a common misconception that full factorial *data collection* is somehow less efficient and not as scalable as fractional factorial. This is *not* true if your subsequent *analysis* looks only at main effects (like most fractional factorial methods). The Google Website Optimizer currently collects data full factorial and analyzes it by looking at main effects only.

If you plan to do only a main effects analysis, the following is true:

- There is no data collection efficiency for fractional over full - each variable value still gets a balanced and proportional portion of the total traffic.
- In the presence of variable interactions (pretty much all of the time in landing page optimization tests) you get less accurate main effects estimates with fractional factorial because you are not considering all of the possible contexts in which a variable is seen.
- You can determine if a main effects analysis is appropriate with full factorial data collection. If it is not, you can do a more complicated analysis by looking at various interactions. With fractional factorial you are stuck with only a main effects model, and you do not even know whether it is a good fit for your data.

In a full factorial parametric test design, the baseline recipe would receive a fraction of the total traffic that is inversely proportional to the size of your search space (e.g., if your search space is 64 recipes, the baseline would receive 1/64th of the total traffic). Since you are trying to beat the existing baseline, special attention should be accorded to it. You also need to be

watching for external and internal changes to your traffic. This means that you want to get accurate conversion information about the baseline recipe even though you may not need to collect a large amount of data for *every* recipe. Because of this, I recommend a modification of the traditional full factorial parametric methodology. As a general rule of thumb, I suggest collecting 15%–25% of your total data on the baseline recipe during a test. This allows you to get tighter error bars on the baseline recipe, and reach statistical significance faster.

Full Factorial Parametric Advantages

There are three main advantages to full factorial parametric tests:

Availability of information on interactions If you use full factorial data collection coupled with a complete model, you can detect all important variable interactions. This is not the case with fractional factorial resolution III designs such as the Taguchi Method, Plackett-Burman, or Latin squares.

Unrestricted test construction You can choose any number of test variables, and arbitrary branching factors for each one. This is in sharp contrast to the significant restrictions found in fractional factorial designs.

Better estimation of main effects Even if you discard the interaction data and only build a model of the main effects, you will still be better off with a full factorial design. If there are interactions, your estimate of the main effects will be more accurate than with fractional factorial designs. This is due to the fact that you have collected data evenly across all recipes (all possible contexts and combinations), and are not relying on spot-sampling a small subset of your search space.

For example, imagine estimating the average elevation of the United States. Full factorial sampling could be compared to sampling on a grid with each measurement a mile apart. A fractional factorial design might be much sparser and might sample on a grid that spaces each measurement every hundred miles. The coarse sampling might overlook geographic features that are smaller than one hundred miles wide, and your elevation estimate might be significantly biased based on the exact position of the sampling grid points. This is much less likely to be a problem with the finer grid, since you would capture all significant features greater than a mile wide, and could not go too far astray in your estimate.

Full Factorial Parametric Disadvantages

There are several disadvantages to full factorial parametric tests:

Very small test size Because of the exponential growth of the number of model coefficients as you increase the number of variables (and/or their branching factors), full factorial design quickly hits its limits *if you are planning to conduct an analysis of all possible interactions*. Because of this, an analysis that includes all possible interactions is rarely used in landing page optimization unless your search space is smaller. However, remember that the search space size can remain as large as a comparable fractional factorial test if you are planning to only model main effects. There is no disadvantage in the speed of data collection under such circumstances.

Complicated analysis Although Google Website Optimizer and other full factorial testing tools are available, most of them will only report on the main effects within your test (the significance of the individual variable values). If you collect information about possible variable interactions

to ensure that you have a more accurate answer, you will have to have a background in statistics to understand which interactions are meaningful.

May not consider variable interactions If you simply conduct a main effects analysis after collecting the data, you will not find variable interactions. In this situation you will also be subject to the piecewise construction errors that I discussed in the "Fractional Factorial Disadvantages" section. However, your estimate of the main effects will be more accurate than fractional factorial.

High uncertainty at the recipe level One other slight drawback of the full factorial parametric approach is that the amount of data that you collect on each individual recipe is small. So you may have poor resolution (a lot of variance) at the recipe level. Because of this, you usually have to run a follow-up test to see if your predicted best answer holds up in the real world. But follow-up tests are also an absolute requirement for all fractional factorial tests, so this is not a relative disadvantage.

10. A real-world example - why fractional factorial is dangerous.

All of my discussion above has been pretty theoretical. But the warnings that I have sounded about fractional factorial approaches cause very severe problems that we repeatedly see in landing page tests. I will use the following real-world example to illustrate the differences between fractional and full factorial data collection and the subsequent data analysis.

SiteTuners' client SF Video handles large-scale DVD duplication and replication. They use a single landing page to drive targeted pay-per-click (PPC) traffic to. The landing page includes a lead form on the left that allows someone to submit a quote request online. The right part of the page includes a number of client logos as credibility enhancers.

In this simple multivariate test we had two variables, each with a branching factor of two (i.e. an original version and one alternative). The variables were as follows:

- V1 - Form headline
 - a - Original - "Free Quote Request"
 - b - New - "Instant Quote"
- V2 - Number of client logos
 - a - Original - 36 logos
 - b - New - 6 logos

The resulting four unique recipes are shown below.

The screenshot shows the SF Video landing page. On the left is a 'Free Quote Request' form with fields for 'Quantity of DVDs' (set to 'Under 1,000'), 'Name', 'Company', 'E-mail Address', 'Phone', and 'Packaging'. A 'Submit' button is at the bottom of the form. On the right, under the heading 'Over 2400 satisfied clients ...', is a grid of 36 logos for various companies and institutions, including Wal-Mart, Microsoft, American Red Cross, Nike, GAP, United Way, ABC, NBC, AT&T, Vivendi Universal, American Lung Association, FOX, Conair, Lexmark, Fisher-Price, Serta, Dreyers, Bell, Touchstone Television, Starz, Tower Records, Herbalife, Solimar, Adams Golf, and others.

Recipe aa - the original version - "Free Quote Request", 36 logos



Recipe *ab* - "Free Quote Request", 6 logos



Recipe *ba* - "Instant Quote", 36 logos



Recipe *bb* - "Instant Quote", 6 logos

The following screenshot of the Google Website Optimizer "Combinations" report shows the statistics that were gathered during the test about each recipe.

Combinations		Page Sections				
Analysis for: Mar 29, 2007 1:04:35 PM PT - Oct 9, 2007 1:50:23 PM PT						
View: <input checked="" type="radio"/> Best 3 Combinations <input type="radio"/> Worst 3 Combinations		Download: Print Preview				
Recipe	Estimated Conversion Rate Range [?]	Chance to Beat Orig. [?]	Chance to Beat All [?]	Observed Improvement [?]	Conversions / Visitors [?]	
aa	2.73% ± 0.6%	—	0.96%	—	38 / 1393	
ba	4.33% ± 0.7%	98.7%	96.3%	58.7%	57 / 1317	
bb	2.86% ± 0.6%	58.4%	1.85%	4.97%	39 / 1362	
ab	2.70% ± 0.6%	48.5%	0.91%	-0.85%	36 / 1331	

As you can see, recipes *aa*, *bb*, and *ab* were basically identical. There was no statistically significant difference among them (their error bars overlap vertically on the graph above). Recipe *ba* was the clear winner with a 98.7% chance of beating the original.

The “Page Sections” report shows the corresponding main effects analysis.

Combinations		Page Sections				
Analysis for: Mar 29, 2007 1:04:35 PM PT - Oct 9, 2007 1:50:23 PM PT						
Sort By: <input checked="" type="radio"/> Relevance Rating <input type="radio"/> Order Created		Download: Print Preview				
Relevance Rating [?]	Variation	Estimated Conversion Rate Range [?]	Chance to Beat Orig. [?]	Chance to Beat All [?]	Observed Improvement [?]	Conversions / Visitors [?]
Heading 2 / 5 	Original	2.72% ± 0.4%	—	3.39%	—	74 / 2724
	Instant Quote	3.58% ± 0.5%	96.5%	96.6%	31.9%	96 / 2679
Images 1 / 5 	Original	3.51% ± 0.5%	—	93.5%	—	95 / 2710
	6 Images	2.78% ± 0.4%	6.55%	6.45%	-20.6%	75 / 2693

Based on this, most people would conclude that the “instant quote” headline is clearly superior, and that the 6-logo version was inferior to the original 36-logo design. If you had only looked at the main effects, you would have concluded that the new headline was the best (96.6% confidence) and that the original client logo layout was superior (93.5% confidence).

But this does not tell the whole story. A closer look reveals that changing the headline and leaving the client logos unchanged had a significant and large impact on the conversion rate. However, there were no big changes in conversion rate with *either* headline in the presence of the new client logo layout.

The SF Video experiment was very simple—in fact you can’t have a simpler multivariate test construction. Why did the headline have a very large impact in the presence of the original client logo layout, while it had none in the presence of the new one? Who knows? The point is that surprisingly large interactions can exist even in such small test designs.

What if we had run this as a fractional factorial test, we would have sampled only recipes *aa*, and *bb*. This would have led us to the conclusion that there was no statistically significant

difference based on either variable in the test. We would have completely missed the powerful interaction between V1a and V2b.

The takeaways are:

- **Fractional factorial would have led to the wrong conclusion** - Since only recipes *aa* and *bb* would have been sampled, the test would have concluded that there was no difference in conversion rate due to either setting of either variable.
- **There is no data collection efficiency for fractional factorial if you are subsequently doing a main-effects-only analysis** - Both data collection methods have sampled evenly across all available variable values.

For fractional factorial - half of the data that you collect will involve V1a (from recipe *aa*), and half will involve V1b (from recipe *bb*). Similarly, half of your data will cover V2a (from recipe *aa*), and half will involve V2b (from recipe *bb*).

For full factorial - half of the data that you collect will involve V1a (a quarter each from recipes *aa* and *ab*), and half will involve V1b (a quarter each from recipes *ba* and *bb*). Similarly, half of your data will cover V2a (a quarter each from recipes *aa* and *ba*), and half will involve V2b (a quarter each from recipes *ab* and *bb*).

- **Full factorial main effects analysis would have led you the right answer for the wrong reason** - If you looked at the piecewise results from the "Page Sections" report, you would have indeed picked recipe *ba*. But you would have done it because V1b was better, and V2a was better. In fact, the combination V1b:V2a has a very strong positive interaction, while the other effects in the model don't matter.
- **Creating a model that included variable interactions after collecting full factorial data is the right approach** - The model with 2-way interactions would have produced the right answer for the right reason. If this was not a toy (2x2) multivariate test, even a full factorial main effects analysis piecewise construction could have easily led to the wrong conclusion. If there are no higher level interactions, you are still free to conduct a main effects analysis, and you will get the correct answer. But there is no way to even know whether interactions exist if you restrict your data collection to fractional factorial.

11. Non-parametric Testing

Non-parametric data analysis can be used with full factorial data collection. In fact, as I mentioned earlier, it *cannot* be used if you impose any restrictions on the recipes allowed during the data collection. So non-parametric analysis implies full factorial data collection.

To my knowledge, there is only one deployed and proven non-parametric approach currently being used for landing page optimization, and I will sketch out its unique features below. But I am confident that other nonparametric methods will be developed in the future to address the obvious limitations of parametric analysis in the setting of landing page optimization.

Epic Sky (EpicSky.com), SiteTuners' parent company, runs large-scale PPC campaigns for clients. A few years ago we also started working for ourselves as a super affiliate. We were driving high-quality PPC traffic to the landing pages provided by each company with whom we had signed up as an affiliate. In many cases, the conversion rate of the landing pages was horrible. This had a direct impact on our affiliate payouts and the scale of programs that we could profitably run. We figured out quickly that if the conversion rate of the landing pages could be

increased, our profits would skyrocket. Out of this self-serving need, we started looking into landing page testing techniques.

Unfortunately, we found that the state of the art at the time was parametric multivariate testing (both full and fractional factorial). Because of the significant limitations of these methods, SiteTuners.com spent over three years developing proprietary math specifically tailored to landing page testing. The result is our proprietary TuningEngine technology.

This approach overcomes the two main limitations common to parametric model building. It takes into account all important variable interactions among your input variables *and* it can scale to very large search space sizes.

What makes this possible is a completely different mathematical framework and approach to the landing page optimization problem. Parametric analysis relies on a model building approach. In other words, it tries to gather enough data to accurately estimate the importance of all variable values to the quantity being measured (i.e., the conversion rate). Once the model is built, it is possible to estimate which variables contributed to the improved performance, and how much they contributed.

By contrast, the TuningEngine asks which combination of variable values (i.e., recipe) is the best-performing one. But it cannot determine which individual variables in the recipe contributed the most to the result.

We have given up the explanatory power in exchange for tangible considerations. Online marketing is a practical discipline grounded in financial and business considerations. Your goal should be to find the landing page that makes you the most money. Explaining why the winning recipe is the best is a secondary academic consideration.

Full Factorial Non-parametric Advantages

The TuningEngine has several advantages over parametric approaches:

Considers variable interactions As I mentioned, the TuningEngine takes variable interactions into account. Although full factorial parametric designs also do this, they are very limited in search space size. Most fractional factorial designs ignore important and prevalent variable interactions altogether.

Handles huge test sizes The TuningEngine can handle very large search spaces. We routinely run tests involving millions or tens of millions of possible recipes. This scale is several orders of magnitude larger than what can be done with multivariate tests (using the same data rate). Unlike other companies in our field, SiteTuners.com does not just collect data on a few dozen recipes at a time. We cover the whole search space. (i.e., our data collection is not extremely sparse, and we can take variable interactions into account).

No experimental design required Most fractional factorial designs require you to have a very specific test construction. This means you may only have a certain number of variables, and their branching factors must also conform to the specifics of the test design matrix being used. This often forces online marketers to limit their creative options or change their test plan in order to conform to these artificial requirements.

The TuningEngine does not place any constraints on your test design. You simply specify the variables that you want to test along with as many alternative values for each one as you

would like. The branching factor for each variable can be different. This is also an advantage for full factorial parametric designs.

Very simple analysis The final phase of a TuningEngine test is a simple A-B split test. The significance of the results is easy to understand (using a single simple statistical test).

Full Factorial Non-parametric Disadvantages

The TuningEngine has three main disadvantages.

Requires higher data rates As a rule of thumb, the TuningEngine requires at least one hundred conversions per day to produce results in a reasonable time frame. This is higher than the minimums recommended for A-B split tests or parametric multivariate tests. Many SiteTuners.com clients actually have multiples or orders of magnitude larger data rates than the minimum, and that is when the technology really begins to shine. However, if you cannot meet the minimum requirement, you may simply have no other option than to use A-B split or parametric multivariate testing.

Cannot tell you why the best landing page works As I discussed earlier, the Tuning- Engine cannot tell you why the winning recipe is the best. In other words, you cannot decompose the results into the contributions of the individual variables. However, this is not as much of a drawback as it first seems. Since variable interactions are prevalent in landing page optimization, any intuition about individual variables is frequently lost anyway. The value of the winning recipe lies in the *combination* of its variable values. So decomposing it into main effects is often a misleading exercise in fantasy.

Proprietary technology Since the TuningEngine is a proprietary "black box", the only way to access it is to hire SiteTuners for a full-service engagement or to license the technology from SiteTuners to run your own engagements.

12. Conclusions

So what is the bottom line?

Among parametric methods, fractional factorial approaches are absolutely the wrong way to go for landing page testing. They do not have any speed advantages during data collection over full factorial. They produce inaccurate or suboptimal results because of their inability to model variable interactions (which are very prevalent and strong in landing page testing).

If you insist on using parametric approaches, you should use full factorial data collection. That way you will have the option of investigating variable interactions during your analysis, and get more accurate estimates even if you subsequently only conduct a main effects analysis.

Non-parametric methods such as the SiteTuners TuningEngine are available for tests with higher data rates. These approaches have none of the drawbacks of fractional factorial. They support much larger test sizes, consider variable interactions, allow unrestricted designs during test construction, and do not require advanced knowledge of statistics.

About SiteTuners.com

SiteTuners.com is a performance-based landing page optimization company. Along with flat-fee engagement pricing, SiteTuners offers a [pure-performance based payment option](#) and gets paid only on the verified profit improvement generated as a result the improved conversion rates. SiteTuners is a charter Google Website Optimizer Authorized Consultant company, and uses it to run A-B Split and full factorial multivariate tests. It also uses its own proprietary non-parametric TuningEngine technology for very large-scale tests. The [TuningEngine can be licensed](#) by interactive agencies and clients who would like to conduct their own testing in-house. SiteTuners specializes in large-scale tests with significant traffic data rates. It works across all industries and tunes a variety of processes including downloads, online registrations, lead forms, and e-commerce.

For smaller businesses that may not have the minimum data rates required for testing, SiteTuners also offers a [Landing Page Conversion Audit](#) consulting service that reviews a landing page to identify issues and suggests best-practices improvements and redesigned page mock-ups.

SiteTuners President Tim Ash is a [recognized authority](#) in the field, contributing expert columnist to [Search Engine Watch](#), a frequent speaker at industry conferences worldwide, and the author of Amazon's e-commerce bestseller *Landing Page Optimization: The Definitive Guide to Testing and Tuning for Conversions* (John Wiley & Sons Press, 2008) LandingPageOptimizationBook.com.

If you have a landing page with over 100 conversion actions per day, please contact us for a [complimentary landing page review](#).

Partial Client List:

