LOOKING FOR THE CLUSTERING EFFECTS OF GOOD OPTIMIZATION

by Sunny J. Harris

A great believer in systematic trading methods, I also adhere to strict testing and optimization guidelines. While we now have access, thanks largely to the ongoing efforts of Omega Research, to powerful research software, we also face the dangers of misunderstanding and misusing the software.

TradeStation 5.0 promises to lift the restrictions on historical testing by removing the 13,000 bar limitation. Having previously tested theories in sections of 13,000 bars, we were inadvertently kept from the full power of optimization. If your system was based on 5-minute bars, you could at most run tests on 166 days worth of data at one time. Testing 15-minute bars allowed you to view 500 days of data. With the limits removed, you could theoretically test the full breadth of available data at one sitting.

At first glance, removing the restriction on the number of bars you can test with TradeStation seems to be a wonderful and much awaited improvement to the software. The potential to generate highly optimized systems, however, looms large. For instance, one could use the data for the S&P 500 contract from 1982 until the present and cleverly produce a system that works wonders in the past and may or may not work at all in the future. It is the subject of proper optimization that I want to discuss in this article.

Every system has certain constants. These constants can always be turned into variables and then optimized. In this article we will investigate a simple moving average crossover system, which has two variables: *length1* and *length2*. The variable *length1* is the length of the slow moving average; the variable *length2* is the length of the fast moving average. It is these two variables we will optimize.

The danger is that most systems created on past data get over-optimized (curve-fitted) to that past data, so that in hindsight the results look terrific. Yet in the real world, tomorrow's optimal parameters are almost always not the same as yesterday's optimal parameters. The system that worked so well on past data falls apart in the real world.

"Optimization is not the enemy—abuse of optimization is." —Ralph Vince, *Portfolio Management Formulas*

"Curve fitting is right when it is used correctly, it's only wrong when it's used incorrectly." —Joe Ross, *Trading by the Book*

According to the dictionary, optimization is the process of finding the best or most favorable conditions or parameters. In system testing, the most favorable conditions are represented by the largest net profit, usually. There are circumstances under which you would not consider the greatest net profit to be the best system. For instance, you might reject a system which has 1000 trades per month and a huge net profit in favor of a system which trades 30 times per month and makes a large, but not the largest, net profit. As you optimize the parameters in your system, you look for variables that produce the greatest net profit over the longest period of time.

Curve-Fitting is the process of developing complicated rules that map known conditions. If you optimize a moving average to loosely fit market conditions from 1982 to 1997, you might do pretty well trading in the future. But if you optimize it so that it very closely fits the market, and you add conditions that match current and past events, you have curve-fitted the data. A curve–fitted system often does not behave well (profitably) into the future.

How much data do you need to use for testing before you can be sure of the replicability of the results? And furthermore, is there a way to keep from curve-fitting the data?

I like to see at least one hundred trades from a system before I am willing to trade it with real money. Most often you will hear the number thirty tossed around, but I am not personally comfortable with so few examples.

To test a system over a set of data which produces only thirty trades would generate a very nice model that would work well over that specific set of historical data. But as the markets fluctuate and the mood changes from bull, to bear, to sideways and back again, the model with thirty trades would probably prove to be only accurate over that subset of data that you initially developed it for. Not only do I like to see 100 trades before I believe a system works, I like the data to include up-moves, down-moves and sideways-moves.

Often I am asked what time period the 100 trades should cover. My response is: as much time as it takes to get 100 trades. If your system generates 100 trades per year, you need a year’s worth of data. If it generates 100 trades in 10 years, you need 10 year’s worth of data.

Testing Your Data by Thirds

The key to successful system generation is to start your research using only a portion of the available data. After completing the entire research process for that limited set of data, engage in “back-testing” and “forward-testing.” Divide your data into thirds, testing the middle third first as you generate theories and systems.

Nine Years of Data Divided into Thirds

1987-1989	1990-1992	1993-1995

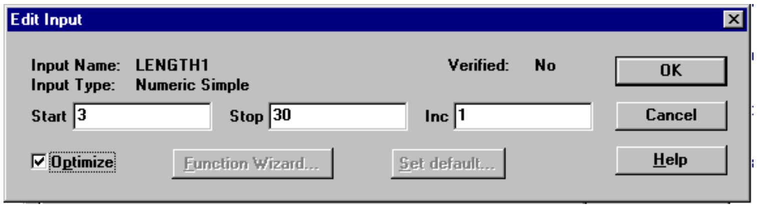
Figure 1

If you were to use the nine years of data from 1987 through 1995 for testing, the data from 1987 through the end of 1989 would be historical data to your new system. Thus, any testing you do on data that happened before the period you've proven is "back-testing." Likewise, the data that comes after 1992 (which you didn't even peek at) is future data to the system. In effect, you are running a blind trial that could not have been curve-fit, because you didn't have the data as part of the initial design.

If your system performs reasonably well (within your expectations) over the backward test and the forward test, you just might have a tradable system. But, don't trade it yet, there's more to find out.

Let's say we were investigating a simple moving average crossover system and wanted to test every possible combination from 3 to 30. That means we want to test every moving average crossover from (3,3), (3,4), (3,5), (3,6) … (3,30), etc. all the way to (30,3), (30,4), (30,5) … (30,30).

Both in TradeStation and SuperCharts you would vary the two lengths by specifying a range of values like this:



Edit Input

Input Name: LENGTH1
Input Type: Numeric Simple
Verified: No

Start 3
Stop 30
Inc 1

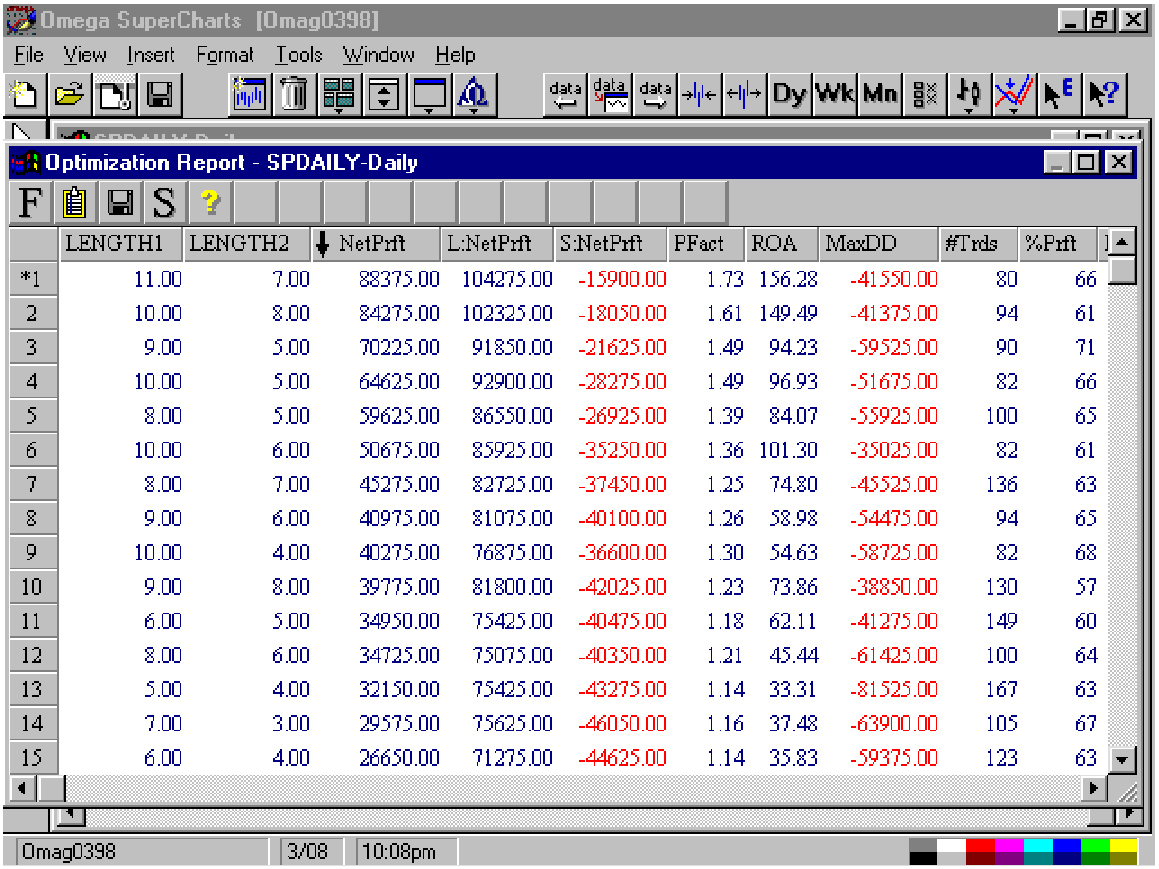
☑ Optimize

Figure 2

Likewise for the second variable, length2, we would input 3 for the start value, 30 for the stop value and 1 for the increment. In shorthand notation this would be expressed as 3:30:1. This combination would cause the software to run 784 tests automatically for us.

When the software finishes running the 784 tests, it displays the results in report format. You will be able to view the net results of all the tests, and compare them to the values used for input. Viewing each parameter that was used as input, and comparing the net profit from each set allows you to determine the optimal input parameters for your eventual system.

Optimization Report from TradeStation or SuperCharts:


Omega SuperCharts [Omag0398]
File View Insert Format Tools Window Help
Optimization Report - SPDAILY-Daily
Omag0398
3/08
10:08pm

	LENGTH1	LENGTH2	NetPrft	L:NetPrft	S:NetPrft	PFact	ROA	MaxDD	#Trds	%Prft
*1	11.00	7.00	88375.00	104275.00	-15900.00	1.73	156.28	-41550.00	80	66
2	10.00	8.00	84275.00	102325.00	-18050.00	1.61	149.49	-41375.00	94	61
3	9.00	5.00	70225.00	91850.00	-21625.00	1.49	94.23	-59525.00	90	71
4	10.00	5.00	64625.00	92900.00	-28275.00	1.49	96.93	-51675.00	82	66
5	8.00	5.00	59625.00	86550.00	-26925.00	1.39	84.07	-55925.00	100	65
6	10.00	6.00	50675.00	85925.00	-35250.00	1.36	101.30	-35025.00	82	61
7	8.00	7.00	45275.00	82725.00	-37450.00	1.25	74.80	-45525.00	136	63
8	9.00	6.00	40975.00	81075.00	-40100.00	1.26	58.98	-54475.00	94	65
9	10.00	4.00	40275.00	76875.00	-36600.00	1.30	54.63	-58725.00	82	68
10	9.00	8.00	39775.00	81800.00	-42025.00	1.23	73.86	-38850.00	130	57
11	6.00	5.00	34950.00	75425.00	-40475.00	1.18	62.11	-41275.00	149	60
12	8.00	6.00	34725.00	75075.00	-40350.00	1.21	45.44	-61425.00	100	64
13	5.00	4.00	32150.00	75425.00	-43275.00	1.14	33.31	-81525.00	167	63
14	7.00	3.00	29575.00	75625.00	-46050.00	1.16	37.48	-63900.00	105	67
15	6.00	4.00	26650.00	71275.00	-44625.00	1.14	35.83	-59375.00	123	63

Figure 3

For the figure above, the report has been sorted by net profit, so you could immediately see which parameters yield the best performance.

If you are observant, you have immediately noticed a problem. We wanted length1 to be the slow moving average and length2 to be the fast moving average. The best net profit in the figure above is generated by a combination of (11,7). That's backwards! You might ask, what's wrong with that? But that subject is for another article. You may either eliminate these erroneous results in your spreadsheet, or you may use EasyLanguage to eliminate them altogether.

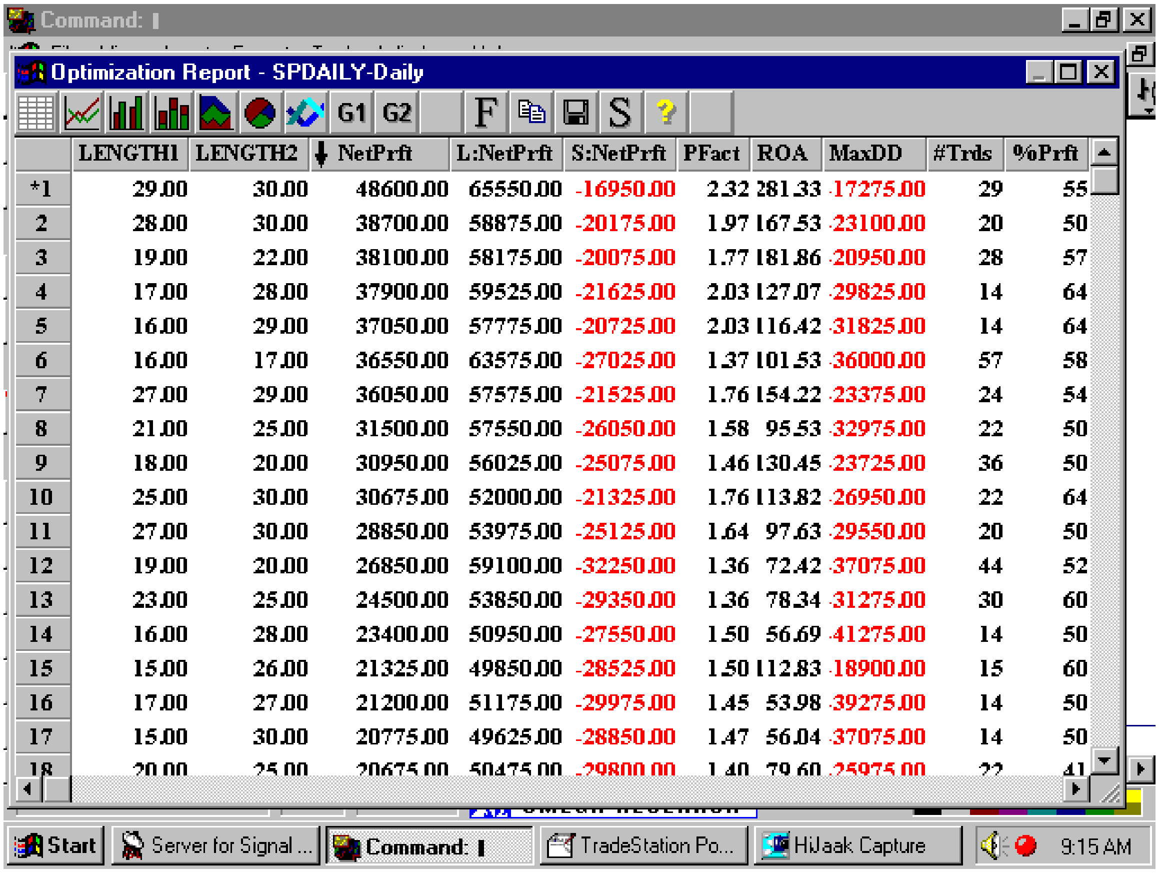
To restrict the moving average crossover system to valid parameters only, simply change the default code from:

CurrentBar > 1 and Average(Close,Length1) crosses over Average(Close,Length2)

to:

CurrentBar > 1 and (Length1<Length2) and Average(Close,Length1) crosses over Average(Close,Length2)

This change will eliminate all the results where the moving averages are equal to each other and where the length of the slow moving average exceeds the length of the fast moving average. The results of the new optimization, again sorted by net profit, follow:



	LENGTH1	LENGTH2	↓ NetPrft	L:NetPrft	S:NetPrft	PFact	ROA	MaxDD	#Trds	%Prft
*1	29.00	30.00	48600.00	65550.00	-16950.00	2.32	281.33	-17275.00	29	55
2	28.00	30.00	38700.00	58875.00	-20175.00	1.97	167.53	-23100.00	20	50
3	19.00	22.00	38100.00	58175.00	-20075.00	1.77	181.86	-20950.00	28	57
4	17.00	28.00	37900.00	59525.00	-21625.00	2.03	127.07	-29825.00	14	64
5	16.00	29.00	37050.00	57775.00	-20725.00	2.03	116.42	-31825.00	14	64
6	16.00	17.00	36550.00	63575.00	-27025.00	1.37	101.53	-36000.00	57	58
7	27.00	29.00	36050.00	57575.00	-21525.00	1.76	154.22	-23375.00	24	54
8	21.00	25.00	31500.00	57550.00	-26050.00	1.58	95.53	-32975.00	22	50
9	18.00	20.00	30950.00	56025.00	-25075.00	1.46	130.45	-23725.00	36	50
10	25.00	30.00	30675.00	52000.00	-21325.00	1.76	113.82	-26950.00	22	64
11	27.00	30.00	28850.00	53975.00	-25125.00	1.64	97.63	-29550.00	20	50
12	19.00	20.00	26850.00	59100.00	-32250.00	1.36	72.42	-37075.00	44	52
13	23.00	25.00	24500.00	53850.00	-29350.00	1.36	78.34	-31275.00	30	60
14	16.00	28.00	23400.00	50950.00	-27550.00	1.50	56.69	-41275.00	14	50
15	15.00	26.00	21325.00	49850.00	-28525.00	1.50	112.83	-18900.00	15	60
16	17.00	27.00	21200.00	51175.00	-29975.00	1.45	53.98	-39275.00	14	50
17	15.00	30.00	20775.00	49625.00	-28850.00	1.47	56.04	-37075.00	14	50
18	20.00	25.00	20675.00	50475.00	-29800.00	1.40	79.60	-25975.00	22	41

Figure 4

Comfort Zones

After you've run thousands of tests, how do you make sense of them? You've produced numerous printed reports, but how do they relate to the system that you are trying to find?

The easiest way to interpret the results is to export the optimization results into a spreadsheet where you can rearrange the rows and columns.

From the System Report in TradeStation, click on the "Trade by Trade" icon . This action will bring up a report showing each trade the system hypothetically made. It is this format you want to export to your spreadsheet program. Clicking on the "Save to Disk" icon , followed by entering a filename and clicking the “Spreadsheet” radio button, will create a text file on your hard disk. Be certain to give each of your files a different name and to specify which fields you want included in the output.

The file you have saved to disk will be in ASCII text format, delimited by commas. Opening the file with your spreadsheet program is then simply a matter of importing the text file. I use Microsoft Excel, so the following illustrations will be from that program. You should be able to do just about the same thing with any spreadsheet.

Figure x shows the initial view of the imported spreadsheet.

Initial Excel Spreadsheet

LENGTH1	LENGTH2	NetPrft	L:NetPrft	S:NetPrft
3	5	3600	51550	-47950
3	9	6950	49125	-42175
4	5	4850	52175	-47325
4	7	1400	49050	-47650
4	9	6250	48950	-42700
5	7	3550	49025	-45475
6	7	10550	55400	-44850
10	25	3875	39725	-35850
12	24	1225	40025	-38800

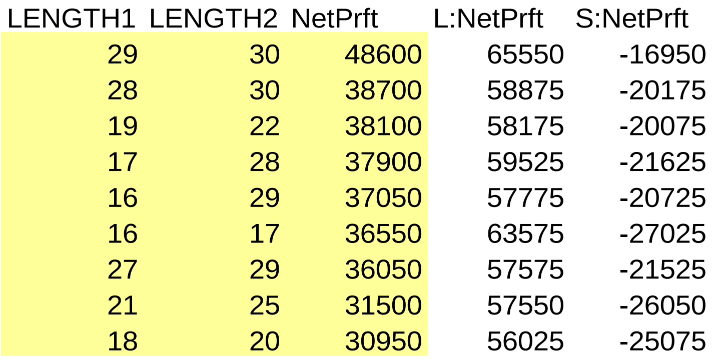
Figure 5

Notice that the initial view, as exported from TradeStation, gives the information in the order of length1 and length2.

Our objective is to see if there is any clustering of parameters that produces a “comfort zone” of good results. If we find that, for instance, dual moving average crossovers of (3,15), (3,16) and (3,17) all produce acceptable net profits, we’ll be much more comfortable trading in this range, than if we find isolated parameters with no clustering.

The first step to finding clusters is to sort the spreadsheet by net profit. That parameter is in column C of our spreadsheet. Let’s set the sort to “descending”, so the largest net profit will be on top.

Spreadsheet Sorted by Net Profits



LENGTH1	LENGTH2	NetPrft	L:NetPrft	S:NetPrft
29	30	48600	65550	-16950
28	30	38700	58875	-20175
19	22	38100	58175	-20075
17	28	37900	59525	-21625
16	29	37050	57775	-20725
16	17	36550	63575	-27025
27	29	36050	57575	-21525
21	25	31500	57550	-26050
18	20	30950	56025	-25075

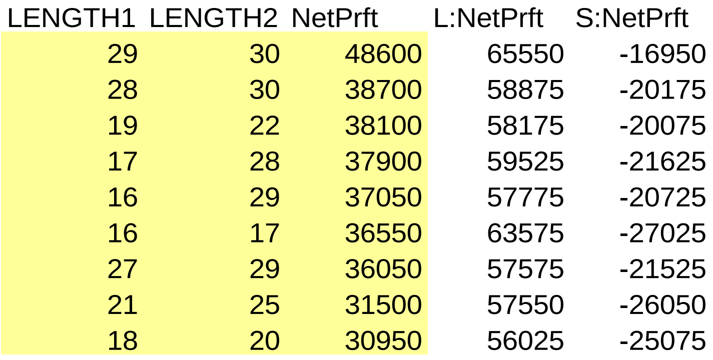
Figure 6

If you are not familiar with these concepts, you have several resources to bring you up to speed on your spreadsheet program. First, the help command in the spreadsheet provides an abundance of information to get you started. Next, you should still have the user's manual for the program. If those two aren't enough, your local community college will probably have classes you can take on a Saturday or in the evening. Further, most bookstores carry video "how to" tapes and "for dummies" books that can get you on your way.

The entire spreadsheet is not included in this article, as it's too large. In the full spreadsheet there are 784 rows for each of the variable sets in the tests. Since we're only interested in the net profit column right now, and in the most profitable results, this subset of the spreadsheet will be adequate.

The second step is to mark the acceptable results so that we will be able to identify them after re-sorting the spreadsheet by columns A and B. I like using color, in particular light yellow, for this purpose. In Excel there is a tool that looks like a paint bucket, which is used to fill the selected cells with your chosen color. Highlight the cells with the most favorable results, down to and including approximately 70% of the maximum net profit. Since 70% of 48600 is 34020, we'll highlight down to the $30,000 range. The result will look like Figure 7.

Re-Sorted Spreadsheet with Best Net Profits Highlighted



LENGTH1	LENGTH2	NetPrft	L:NetPrft	S:NetPrft
29	30	48600	65550	-16950
28	30	38700	58875	-20175
19	22	38100	58175	-20075
17	28	37900	59525	-21625
16	29	37050	57775	-20725
16	17	36550	63575	-27025
27	29	36050	57575	-21525
21	25	31500	57550	-26050
18	20	30950	56025	-25075

Figure 7

Now when we sort the spreadsheet by columns A and B, which contain the data for length1 and length2, the highlighted cells will be easy to spot, as in Figure 8.

Spreadsheet Sorted by Length1 and Length2

LENGTH1	LENGTH2	NetPrft	L:NetPrft	S:NetPrft
29	30	48600	65550	-16950
28	29	18125	48100	-29975
28	30	38700	58875	-20175
27	28	17725	48400	-30675
27	29	36050	57575	-21525
27	30	28850	53975	-25125
26	27	10700	42000	-31300
26	28	2725	38025	-35300
26	29	7225	40275	-33050
26	30	15450	44400	-28950
25	27	6850	41350	-34500
25	30	30675	52000	-21325
24	25	14750	47925	-33175
24	30	17925	46900	-28975
23	24	3750	43475	-39725
23	25	24500	53850	-29350
23	30	9075	45200	-36125

Figure 8

There will be several clusters highlighted yellow. As you run your tests on more and more data, you will begin to notice cluster areas where there are overlaps in good performance from month to month, quarter to quarter, or from year to year. You will want to concentrate your testing and design efforts in these areas, where the predominance of clustering lies.

If you trade in an out-lying area, just because it has the most profit, you will surely experience the pitfalls of over-optimization and quickly enter into severe drawdown. Don’t do it. Stay within the middle of a wide colored area.