CHAPTER 9

The Leverage Space Model

Since the 1950s, when formal portfolio construction was put forth, people have sought to discern optimal portfolios as a function of two competing entities, risk and return. The objective was to maximize return and minimize risk. This is the old paradigm. It's how we have been taught to think.

Quoting from Kuhn,[1] "Acquisition of a paradigm and of the more esoteric type of research it permits is a sign of maturity in the development of any given scientific field."

This is precisely what happened. Portfolio construction, after the second world war, acquired a mathematical rigor that had been missing prior thereto. Earlier, it was, as in so many other fields, the fact-gathering phase where each bit of data seemed equally relevant. However, with the paradigm presented as the so-called *Modern Portfolio Theory* (a.k.a. *E-V Theory or mean-variance model*), the more esoteric type of research emerged.

Particularly troubling with this earlier paradigm was the fact that the unwanted entity, risk, was never adequately defined. Initially, it was argued that risk was the variance in returns. Later, as the arguments that the variance in returns may be infinite or undefined, and that the dispersion in returns wasn't really risk, calamitous loss was risk, the definitions of risk became ever more muddled.

Overcoming ignorance often requires a new and different way of looking at things.

---

[1]Thomas S. Kuhn, *The Structure of Scientific Reduction*, The University of Chicago Press, 1962.

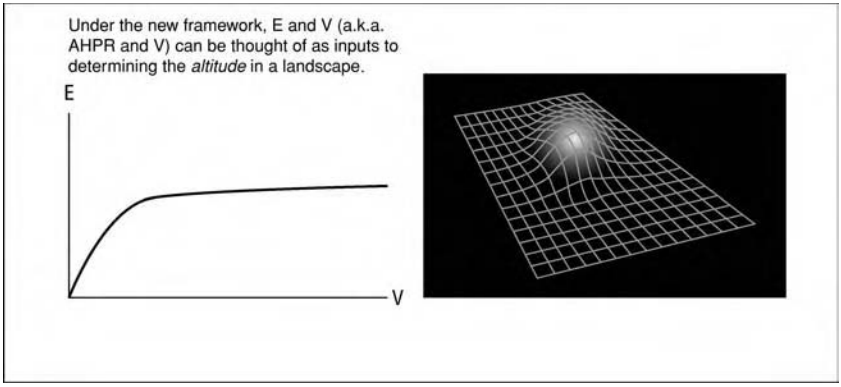

FIGURE 9.1 Conceptual view of the old framework, *left*, with the new, *right*

WHY THIS NEW FRAMEWORK IS BETTER

For nearly four decades, portfolio construction was envisioned in a two-dimensional plane, where return made up the vertical axis, and risk—actually, some surrogate measure of risk—was the horizontal axis. The basic notion was to get as great a return for a given level of risk, or as low a level of risk for a given level of return, as was possible on this two-dimensional plane (see Figure 9.1).

The new framework to be presented is an altogether new way of viewing portfolio construction, different than looking at portfolios in a two-dimensional, risk-competing-with-return sense.[2] There are a number of reasons to opt for the new framework over the old.

The new approach is superior because the inputs are no longer along the lines of expected returns and (the rather nebulous) variance in expected returns, or some other ersatz measure of risk. The inputs to this new model are different *scenarios* of different outcomes that the investments may take (a more accurate approximation for the real distribution of returns). Now, rather than estimating things like expected returns and variance in those expected returns, the inputs are much closer to what the investment manager may be thinking (e.g., a y % chance of an x % gain or loss, etc.). Now,

[2]In Chapter 12 we will see, however, how to take the portfolio constructed from the methods outlined in this chapter, juxtaposed to its respective drawdown and thus truly maximize return for a given level of "risk."

the investment manager can even account for the *far-out*, slim-probability scenarios as inputs to the new model.

What the investment manager uses as inputs to the new model are *spectrums* of scenarios for each market or market system (a given market traded with a given approach). The new model discerns optimal allocations to each scenario spectrum based on trading multiple, simultaneously traded scenario spectrums.

As we have seen, the old model not only used returns and variance in those returns as inputs to the model, but also used the correlation coefficients of the pairwise combinations of those streams of returns.

This last parameter, the *correlation coefficients of the pairwise combinations of the streams in returns*, is critical. Consider again the case of our two-to-one coin toss. If we are playing that particular game alone, our optimal f is .25.

If, however, we play a second and simultaneous game (and, for the sake of simplicity, we say it is the same game—a second two-to-one coin toss), the optimal f values now become a function of the correlation between those two games.

If the correlation is +1.0, we can show that, optimally, we bet $(.25 - x)$ on one game, and x on the other (where $x >= 0$ and $x <= .25$). Thus, in total, when the correlation is 1.0, we never have more than optimal f exposure in total (i.e., in cases of perfect, positive correlation, the total exposure does not exceed the exposure of the single game).

If the correlation were −1.0, the optimal f then goes to .5 for each game, for a net exposure of 1.0 (100%) since, at such a value of the correlation coefficient, a losing sequence of such simultaneous games is impossible for even one play.

If the correlation is zero, we can determine that the optimal bet size between these two games now is .23 on each game, for a total exposure of .46 per play. Note that this exceeds the total exposure of .25 for a single game. Interestingly, when one manages the bankroll for optimal growth, *diversification* clearly does *not* reduce risk; rather, it increases it, as evident here if the one-in-four chance of both simultaneous plays were to go against the bettor, a 46% drawdown on equity would immediately occur.

Typically, correlations as they approach zero only see the optimal f buffered by a small fraction, as evidenced in this illustration of two simultaneously played two-in-one coin tosses.

Here, we are measuring the correlation of binomially distributed outcomes (heads or tails), and the outcomes are random, not generated by human emotions. In other types of environments, such as market prices, correlation coefficients begin to exhibit a very dangerous characteristic.

When a large move occurs in one component of the pairwise combination, there is a tendency for correlation to increase, often very dramatically.

Additionally, since I am speaking here of, say, market A making the large move, and its correlation to B, then too can I expect A and C to see an increase in their correlation coefficient in those time periods of the large move, and hence between B and C during those periods where I see a large move in A.

In short, when the big moves come, things tend to line up and move together (to a far greater degree than the correlation coefficient implies). In incidental time periods, which are most time periods, the correlation coefficients tend back toward zero.

To see this, consider the following study. Here, I tried to choose random and disparate markets. Surely, everyone may have picked a different basket than the random one I drew here, but this basket will illustrate the effect as well as any other. I took three commodities—crude oil (CL), gold (GC), and corn (C)—using continuous back-adjusted contracts, the use of which I devised while working with Bruce Babcock in 1985. I also put in the S&P 500 Cash Stock Index (SPX) and the prices of four individual stocks, Exxon (XOM), Ford (F), Microsoft (MSFT), and Pfizer (PFE). The data used were from the beginning of January 1986 through May 2006—nearly 20 years.

I used daily data, which required some alignment for days where some exchanges were closed and others were not. Particularly troublesome here was the mid-September 2001 period.

However, despite this unavoidable slop (which, ultimately, has little bearing on these results), the study bears out this dangerous characteristic of using correlation coefficients for market-traded pairwise price sequences.

Each market was reduced to a daily percentage of the previous day merely by converting the daily prices for each day as divided by the price of that item on the previous day. Afterward, for each market, I calculated the standard deviation in these daily price percentage changes.

Taking these eight different markets, I first ran their correlation coefficients over the daily percentage price data in question. This is shown in the "All days," section, and *is* the benchmark, as it is typically what *would* be used in constructing the classical portfolio of these components.

Next, I took each component and ran a study wherein the correlations of all components in the portfolio were looked at, but only on those days where the distinguishing component moved beyond 3 standard deviations that day. This was also done for days where the distinguishing component moved less than one standard deviation that day (the "Incidental days").

This can be seen as follows. The section titled "CL beyond 3 sigma" shows the correlation of all components in the study period on those days where crude oil had a move in excess of 3 standard deviations.

Similarly, the section that follows, where we see "CL within 1 sigma," shows the correlation of all components in the study period on those days where crude oil had a move of less than 1 standard deviation.

Consider now the correlation for crude oil and gold, which shows for "All days" as 0.18168298092886612. When crude oil has had a move in excess of 3 standard deviations, gold has moved much more lockstep in the same direction, now exhibiting a correlation of 0.6060715468257946.

On those more "Incidental days," where crude oil has moved less than 1 standard deviation, gold has moved nearly randomly with respect to it, now showing a correlation coefficient of 0.08754532513257751.

Of note on the method of calculation used in determining the means of the percentage price changes, which are used to discern standard deviations in the percentage price changes, as well as the standard deviations themselves, I did *not* calculate these simply over the entire data set. To do so would have been to have committed the error of perfect foreknowledge. Rather, at each date through the chronology of the data used, the means and standard deviations were calculated only up to that date, as a rolling 200-day window. Thus, I calculated rolling 200-day standard deviations so as to avoid the fore-knowledge trap. Thus, the actual starting date, after the 200-day required data buildup period, was (ironically) October 19, 1987 (and therefore yields a total of 4,682 trading days in this study).

This study is replete with example after example of this effect of large moves in one market portending corresponding large moves in other markets, and vice versa. As the effect of correlation is magnified, the conditions become more extreme For example, look at Ford (F) and Pfizer (PFE). On all days, the correlation between these two stocks is 0.15208857952056634, yet, when the S&P 500 Index (SPX) moves greater than 3 standard deviations, the Ford-Pfizer correlation becomes 0.7466939906546621. On days where the S&P 500 Index moves less than 1 standard deviation, the correlation between Ford and Pfizer shrinks to a mere 0.0253249911811074.

Take a look at corn (C) and Microsoft (MSFT). On all days the correlation in the study between these two disparate, tradable items was 0.022097632770092066. Yet, when gold (GC) moved more than 3 standard deviations, the correlation between corn and Microsoft rose to 0.24606355445287773. When gold was within 1 standard deviation, this shrinks to 0.011571945077398543.

Sometimes, the exaggeration occurs in a negative sense. Consider gold and the S&P. On all days, the correlation is −0.140572093416518. On days where crude oil moves more than 3 standard deviations, this rises to −0.49033570418986916, and when crude oil's move is less than 1 standard deviation, it retracts in to −0.10905863263068859.

All days
(4682 of 4682 data points)

CL	GC	0.18168298092886612
CL	C	0.06008614529554469
CL	SPX	−0.06337343876830624
CL	XOM	0.12237528928675677
CL	F	−0.056071166990844516
CL	MSFT	−0.008336837297919815
CL	PFE	−0.03971512674407262
GC	C	0.07558861340485105
GC	SPX	−0.140572093416518
GC	XOM	−0.03185944850989464
GC	F	−0.07649165457662757
GC	MSFT	−0.06175684105762799
GC	PFE	−0.06573632473755334
C	SPX	0.03147493683616401
C	XOM	0.02623205260520187
C	F	0.030704335620653868
C	MSFT	0.022097632770092066
C	PFE	0.013735926438934488
SPX	XOM	0.4463700373245729
SPX	F	0.44747978695133384
SPX	MSFT	0.4644715701985205
SPX	PFE	0.39712431335046133
XOM	F	0.18406887477828698
XOM	MSFT	0.17555859825807965
XOM	PFE	0.17985680973424692
F	MSFT	0.19472214174383298
F	PFE	0.15208857952056634
MSFT	PFE	0.15655275607502264

CL beyond 3 sigma
(57 of 4682 data points)

CL	GC	0.6060715468257946
CL	C	0.16773966461586043
CL	SPX	−0.4889254290079874
CL	XOM	0.30834231052418093
CL	F	−0.4057990096591226
CL	MSFT	−0.043298612614148003
CL	PFE	−0.2862619588205237
GC	C	0.2136979555796156
GC	SPX	−0.49033570418986916
GC	XOM	−0.04638590060660794
GC	F	−0.34101700944373253
GC	MSFT	−0.04792818652129692
GC	PFE	−0.23339206379967778
C	SPX	−0.13498070111097166
C	XOM	0.1282166452534864
C	F	−0.07574638268565898
C	MSFT	−0.046367278697754616
C	PFE	0.02171787217124139
SPX	XOM	0.3720220077411345
SPX	F	0.7508447148878216
SPX	MSFT	0.26583237333985554
SPX	PFE	0.5576012125272648
XOM	F	0.19597328384286486
XOM	MSFT	0.2817265916572091
XOM	PFE	0.14847216371343516
F	MSFT	0.24795671036100472
F	PFE	0.45818973137924285
MSFT	PFE	0.09703388355674258

CL within 1 sigma
(3355 of 4682 data points)

CL	GC	0.08754532513257751
CL	C	0.0257566754226136
CL	SPX	0.018864830486201915
CL	XOM	0.07275446285160611
CL	F	−0.006035919250607675
CL	MSFT	0.0039040541983706815
CL	PFE	−6.725739893499835E-4
GC	C	0.07071392644936346
GC	SPX	−0.10905863263068859
GC	XOM	−0.038050306091619565
GC	F	−0.046995783946869804
GC	MSFT	−0.035463714683264834
GC	PFE	−0.06020481387795751
C	SPX	0.028262511037748024
C	XOM	0.017421211262930312
C	F	0.027058713971227104
C	MSFT	0.023756786611237552
C	PFE	0.014823926818879715
SPX	XOM	0.41388474915130574
SPX	F	0.4175520920293062
SPX	MSFT	0.4157760485443937
SPX	PFE	0.36192135400550934
XOM	F	0.16278071355175439
XOM	MSFT	0.1319530034838986
XOM	PFE	0.1477015704953524
F	MSFT	0.16753417657993877
F	PFE	0.12522622923381158
MSFT	PFE	0.12969188109495833

GC beyond 3 sigma
(49 of 4682 data points)

CL	GC	0.37610799881628454
CL	C	−0.013505453061135679
CL	SPX	−0.4663766105812081
CL	XOM	−0.1236757784439896
CL	F	−0.26893323996770363
CL	MSFT	−0.25074947066586095
CL	PFE	−0.34522609666192644
GC	C	0.12339691398398928
GC	SPX	−0.2256870226039319
GC	XOM	−0.17825193598720657
GC	F	−0.2932885892847866
GC	MSFT	−0.0942827495583651
GC	PFE	−0.08178972441698702
C	SPX	0.2589426127779489
C	XOM	0.324334753787739
C	F	0.17993600277237867
C	MSFT	0.24606355445287773
C	PFE	0.0632678902662783
SPX	XOM	0.6106538927488477
SPX	F	0.7418500480107237
SPX	MSFT	0.814073269082298
SPX	PFE	0.6333158417738232
XOM	F	0.3731941584747982
XOM	MSFT	0.29680898662233957
XOM	PFE	0.5191106683884512
F	MSFT	0.5875623837594202
F	PFE	0.35514526049741935
MSFT	PFE	0.46225966739620467

C beyond 3 sigma
(63 of 4682 data points)

CL	GC	0.09340139862063926
CL	C	0.15937424801870365
CL	SPX	−0.034836945862889324
CL	XOM	0.31262202861570143
CL	F	−0.0015035928633431528
CL	MSFT	−0.035100428463551
CL	PFE	−0.042790208990554315
GC	C	−0.07554730971707264
GC	SPX	−0.09770624459871546
GC	XOM	−0.1178996789974603
GC	F	−0.1580599457490364
GC	MSFT	−0.017408456343824652
GC	PFE	−0.057116412345416667
C	SPX	−0.12610050901450232
C	XOM	−0.06491379177062588
C	F	0.13713180201552985
C	MSFT	0.1184669909561641
C	PFE	0.07365117745748967
SPX	XOM	0.6379868873961733
SPX	F	0.6386287499447472
SPX	MSFT	0.3141265015844073
SPX	PFE	0.07148466884745952
XOM	F	0.352541750183325
XOM	MSFT	0.15822517152455984
XOM	PFE	−0.01714503647656309
F	MSFT	0.2515504291514764
F	PFE	−0.17915715988166248
MSFT	PFE	4.0302517044280364E-4

GC within 1 sigma
(3413 of 4682 data points)

CL	GC	0.08685001387886367
CL	C	0.03626120508953206
CL	SPX	−0.026042510508209223
CL	XOM	0.12444488722949365
CL	F	−0.03218089855875674
CL	MSFT	−0.0015484284736459364
CL	PFE	−0.023185426431743598
GC	C	0.036165047559364234
GC	SPX	−0.1187633862400288
GC	XOM	−4.506758967026326E-5
GC	F	−0.05680170397975439
GC	MSFT	−0.04749027255821666
GC	PFE	−0.05546821106288489
C	SPX	0.020548509330959506
C	XOM	0.009891493444709805
C	F	0.03164457405193553
C	MSFT	0.011571945077398543
C	PFE	0.021658621577528698
SPX	XOM	0.38127728674269895
SPX	F	0.45590091052598297
SPX	MSFT	0.4658428532832456
SPX	PFE	0.34733314433363616
XOM	F	0.15700577420431003
XOM	MSFT	0.12789055576102093
XOM	PFE	0.1226203887798495
F	MSFT	0.19737706075000538
F	PFE	0.11755272888079606
MSFT	PFE	0.13784745249948008

C within 1 sigma
(3391 of 4682 data points)

CL	GC	0.17533527416024455
CL	C	0.026858830610224073
CL	SPX	−0.0732811159519982
CL	XOM	0.1028138088787534
CL	F	−0.05102926721840804
CL	MSFT	−0.01099110090227016
CL	PFE	−0.047128710608280625
GC	C	0.05773910871663286
GC	SPX	−0.1360779110437837
GC	XOM	−0.02099718827227882
GC	F	−0.06222113210658744
GC	MSFT	−0.04966940059247658
GC	PFE	−0.07413097933730392
C	SPX	−0.00883286682481027
C	XOM	−4.4357501736777734E-4
C	F	−0.003482794137395384
C	MSFT	0.0011277030286577093
C	PFE	0.006559218632362692
SPX	XOM	0.3825048808789464
SPX	F	0.41829697072918165
SPX	MSFT	0.4395087414084105
SPX	PFE	0.49804329260547564
XOM	F	0.1475733885968429
XOM	MSFT	0.13663720618579042
XOM	PFE	0.21209220175136173
F	MSFT	0.16502841838609542
F	PFE	0.188267473055017
MSFT	PFE	0.1868337356456869

SPX beyond 3 sigma
(37 of 4682 data points)

CL	GC	0.262180235243967
CL	C	0.2282732831599413
CL	SPX	0.09510759900263809
CL	XOM	0.15585802115704978
CL	F	0.03830267479460007
CL	MSFT	0.11346892107581757
CL	PFE	0.014716269207474146
GC	C	−0.2149326327219606
GC	SPX	−0.2724333717672031
GC	XOM	−0.20973685485328555
GC	F	−0.5133205870466547
GC	MSFT	−0.2718742251789026
GC	PFE	−0.15372156278838536
C	SPX	0.27252943570443455
C	XOM	0.28696147861064464
C	F	0.28903764586090686
C	MSFT	0.2682496194114376
C	PFE	0.1575739360953595
SPX	XOM	0.8804915455367398
SPX	F	0.8854422072373676
SPX	MSFT	0.9353021184213065
SPX	PFE	0.8785677290825313
XOM	F	0.7720878305603963
XOM	MSFT	0.8107472671261666
XOM	PFE	0.8581109151100405
F	MSFT	0.867848932613579
F	PFE	0.7466939906546621
MSFT	PFE	0.8244864622745551

XOM beyond 3 sigma
(31 of 4682 data points)

CL	GC	0.08619386913767751
CL	C	0.12281769759782755
CL	SPX	0.1598136682243572
CL	XOM	0.19657554427842094
CL	F	0.20764047880440853
CL	MSFT	0.20143983941373977
CL	PFE	0.06491145921791507
GC	C	−0.3440263176542505
GC	SPX	−0.6127703828515739
GC	XOM	−0.21647163055987845
GC	F	−0.5586655697340519
GC	MSFT	−0.49757437569583096
GC	PFE	−0.6574499556463053
C	SPX	0.46950837936435447
C	XOM	0.10204725109291456
C	F	0.5528812200193067
C	MSFT	0.3962060773300878
C	PFE	0.4835629447364572
SPX	XOM	0.26560300433620926
SPX	F	0.9513940647043279
SPX	MSFT	0.951627088342409
SPX	PFE	0.939838119184664
XOM	F	0.2073529344817686
XOM	MSFT	0.23527599847538386
XOM	PFE	0.1587269337304879
F	MSFT	0.9093988443935644
F	PFE	0.8974023710639419
MSFT	PFE	0.8661556879321936

SPX within 1 sigma
(3366 of 4682 data points)

CL	GC	0.1411703426148108
CL	C	0.07065326135001565
CL	SPX	−0.04672042595452156
CL	XOM	0.1369231929185177
CL	F	−0.03833898351928496
CL	MSFT	0.008249795822319618
CL	PFE	−0.039824997750446386
GC	C	0.07487815673746215
GC	SPX	−0.098702234833124
GC	XOM	0.0126749627781548
GC	F	−0.025504778030182328
GC	MSFT	−0.007650115919919071
GC	PFE	−0.03409826874750128
C	SPX	−0.0037085243318329152
C	XOM	0.007681382976920977
C	F	0.012302593393623804
C	MSFT	0.023440459199345766
C	PFE	0.020051710510815043
SPX	XOM	0.24274905226797128
SPX	F	0.25706355236368167
SPX	MSFT	0.23491561078843676
SPX	PFE	0.22050509324437187
XOM	F	0.051567190213371944
XOM	MSFT	0.011930867235937883
XOM	PFE	0.03903218211997973
F	MSFT	0.049167377717242194
F	PFE	0.0253249911811074
MSFT	PFE	0.01813554953465995

XOM within 1 sigma
(3469 of 4682 data points)

CL	GC	0.1626123169907851
CL	C	0.06385666453921195
CL	SPX	−0.10197617432497605
CL	XOM	0.10671051194661867
CL	F	−0.06561037074518512
CL	MSFT	−0.03369575980606431
CL	PFE	−0.0497046013203275 16
GC	C	0.0699568184904768
GC	SPX	−0.14448139178331096
GC	XOM	−0.02183888080921421
GC	F	−0.07949839243937246
GC	MSFT	−0.06427915157699021
GC	PFE	−0.056426779255276956
C	SPX	0.002666843180930068
C	XOM	0.008152806548151075
C	F	0.02130372788477299
C	MSFT	0.02696846819596459
C	PFE	0.023479323154123974
SPX	XOM	0.4439456452926861
SPX	F	0.410255598243555
SPX	MSFT	0.40962971140985116
SPX	PFE	0.3337542998608116
XOM	F	0.16171670346660708
XOM	MSFT	0.1522471847121916
XOM	PFE	0.14027113549516057
F	MSFT	0.15954186850809635
F	PFE	0.09692360471545824
MSFT	PFE	0.11103574324620878

F beyond 3 sigma
(43 of 4682 data points)

CL	GC	0.27427702981787166
CL	C	−0.036710270159938795
CL	SPX	−0.05122250042406012
CL	XOM	0.019879344178947128
CL	F	−0.1619398623288661
CL	MSFT	0.06113040620102775
CL	PFE	−0.03052373880511025
GC	C	−0.2105245502328284
GC	SPX	−0.39275282180603993
GC	XOM	−0.2660521070959948
GC	F	−0.07998977703405707
GC	MSFT	−0.39045981709259187
GC	PFE	−0.15655811237828485
C	SPX	0.4394625985396639
C	XOM	0.5111084269242103
C	F	0.05517927015323412
C	MSFT	0.418713605628322
C	PFE	0.4114006944120061
SPX	XOM	0.8858315365005958
SPX	F	0.32710966702049354
SPX	MSFT	0.9438851500634157
SPX	PFE	0.842765820623699
XOM	F	0.23769276790825533
XOM	MSFT	0.8786892436047334
XOM	PFE	0.7950187695417785
F	MSFT	0.26860165851836737
F	PFE	0.2978173791782456
MSFT	PFE	0.8111631403849762

MSFT beyond 3 sigma
(39 of 4682 data points)

CL	GC	0.05288220924874525
CL	C	0.03238866347529909
CL	SPX	0.23409424184528582
CL	XOM	0.27655163811605127
CL	F	0.21291573296289484
CL	MSFT	0.2347395935937538
CL	PFE	0.22620918949312924
GC	C	0.17132011394477453
GC	SPX	−0.27621216630360723
GC	XOM	−0.31742556492355695
GC	F	−0.39376436665709946
GC	MSFT	0.03872797470182633
GC	PFE	−0.34653065475607997
C	SPX	0.2841344985841967
C	XOM	0.2722771622858543
C	F	0.1930254456039821
C	MSFT	0.10837798889022507
C	PFE	0.24059844829500385
SPX	XOM	0.9370778598925431
SPX	F	0.9173970725342884
SPX	MSFT	0.21910290988946773
SPX	PFE	0.8750562187811304
XOM	F	0.852903525597108
XOM	MSFT	0.28329029115636173
XOM	PFE	0.8689912705869133
F	MSFT	0.1224603844278996
F	PFE	0.7914349481572399
MSFT	PFE	0.08342580014726039

F within 1 sigma
(3513 of 4682 data points)

CL	GC	0.14512911921800759
CL	C	0.047640657886711776
CL	SPX	−0.038662740379307635
CL	XOM	0.13475499739302577
CL	F	−0.02779741081029594
CL	MSFT	0.002124836307259393
CL	PFE	−0.0346544213095382
GC	C	0.07406272080516503
GC	SPX	−0.08216193364828302
GC	XOM	0.0018927626451161
GC	F	−0.04189153921839398
GC	MSFT	−0.017773478113621854
GC	PFE	−0.03394532760699087
C	SPX	0.00863250682585783
C	XOM	−0.0024652908939917476
C	F	0.03824383087240428
C	MSFT	0.026328712743665918
C	PFE	−0.009582466225759407
SPX	XOM	0.3300910692705658
SPX	F	0.3879282004829515
SPX	MSFT	0.37619527832248406
SPX	PFE	0.3522133339947073
XOM	F	0.12461137390050991
XOM	MSFT	0.08511094562657419
XOM	PFE	0.11899749055724199
F	MSFT	0.1291334261723857
F	PFE	0.09432105016323611
MSFT	PFE	0.10326939903567782

MSFT within 1 sigma
(3788 of 4682 data points)

CL	GC	0.1780064461248614
CL	C	0.05816017421928696
CL	SPX	−0.08387058206522074
CL	XOM	0.11404112460697703
CL	F	−0.0581086900122653
CL	MSFT	−0.04785934015162996
CL	PFE	−0.04252837463155788
GC	C	0.06971353618749605
GC	SPX	−0.10854537254629587
GC	XOM	−0.02305369375053341
GC	F	−0.0433322968281354
GC	MSFT	−0.05714331580093729
GC	PFE	−0.04492680308546143
C	SPX	0.01597033368734557
C	XOM	0.01678577953312174
C	F	0.019585474298717553
C	MSFT	0.021226325810089326
C	PFE	0.01121828967048508
SPX	XOM	0.35173508501967765
SPX	F	0.3788577061068169
SPX	MSFT	0.510722761985027
SPX	PFE	0.3308252244568856
XOM	F	0.12245205070590215
XOM	MSFT	0.11855012193953615
XOM	PFE	0.1127871934860319
F	MSFT	0.18490175993452032
F	PFE	0.1035829207843917
MSFT	PFE	0.16958846505571112

The point is evident throughout this study: Big moves in one market amplify the correlation between other markets, and vice versa. Some explanations can be offered to partially account for this tendency; for one, these markets are all USD denominated, yet, these elements can only partially account as the *cause* of this. Regardless of its cause, even the fact that this characteristic exists warns us that the correlation parameter fails us at those very times when we are counting on it the most.

What we are working with in using correlation is a composite of the incidental time periods and time periods with considerably more volatility and movement. Clearly, it is misleading to use the correlation coefficient as a single parameter for the joint movement of pairwise components.

Additionally, considering that in a normal distribution, 68.26894921371% of the data points will fall within one sigma either side of the mean. Given 4,682 data points, we would expect therefore to typically have 3196.352 data points be within one sigma. But we repeatedly see more than that. We would also expect, given the Normal distribution, for 99.73002039367% of the data points to be within three sigma, thus, $1 - .99730020393 = 0.002699797$ probability of being beyond three sigma. Given 4,682 data points, we would therefore expect $4{,}682 * 0.002699797 = 12.64045$ data points to be beyond three sigma. Yet again, we see far more than this in every case, in every market in this study. These findings are consistent with the "fat tails," notion of price distributions.

If more data points than expected fall within one sigma, and more than expected fall outside of three sigma, then the shortfall must be made up with fewer data points than would be expected between |1| and |2| sigma. What is germane to the discussion here, however, is that days when correlations tend more toward randomness occur far more frequently than would be expected if prices were normally distributed, but, in a manner fatal to the conventional models, the critical days where things move more lockstep occur far more often as well.

Consider again our simultaneous two-to-one coin toss example. We have seen that at a correlation coefficient of zero, we optimally bet .23 on each component. Yet, what if we later learned we were deluded about that correlation coefficient, that, rather than being zero, it was, instead +1.0?

In such a circumstance we would have been betting .46 per play, where the optimal was .25. In short, we would have been far to the right of the peak of the *f* curve.

By relying on the correlation coefficient alone, we delude ourselves. The new model disregards correlation as a solitary parameter of pairwise component movement. Rather, the new model addresses this principle as it *must* be addressed. We are concerned in the new model with the joint probabilities of two scenarios occurring, one from each of the pairwise components, simultaneously, as the history of price data dictates we do.

Furthermore, and perhaps far more importantly, the new model holds for any distribution of returns! The earlier portfolio models most often assumed a normal distribution in estimating the various outcomes the investments may have realized. Thus, the tails—the very positive or very negative outcomes—were much thinner than they would be in a non-normal, real-world distribution. That is, the very good and very bad outcomes that investments can witness tended to be underaccounted for in the earlier models. With the new model, various scenarios comprise the tails of the distribution of outcomes, and you can assign them any probability you wish. Even the mysterious Stable Paretian Distribution of returns can be characterized by various scenarios, and an optimal portfolio discerned from such. Any distribution can be modeled as a scenario spectrum; scenario spectrums can assume any probability density shape desired, and they are easy to do. You needn't ask yourself, "What is the probability of being x distance from the mode of this distribution?" but rather, "What is the probability of these scenarios occurring?"

So the new framework can be applied to any distribution of returns, not simply the normal. Thus, the real-world *fat-tails* distribution can be utilized, as a scenario spectrum is another way of drawing a distribution.

Most importantly, the new framework, unlike its predecessors, is not one so much of composition but rather of progression. It is about leverage, and it is also about how you progress your quantity through time, as the equity in the account changes.

Interestingly, *these are different manifestations of the same thing. That is, leverage (how much you borrow), and how you progress your quantity through time are really the same thing.*

Typically, leverage is thought of as "How much do I borrow to own a certain asset?" For example, if I want to own 100 shares of XYZ Corporation, and it costs $50 a share, then it costs $5,000 for 100 shares. Thus, if I have less than $5,000 in my account, how many shares should I put on? This is the conventional notion of leverage.

But leverage also applies to borrowing your own money. Let's suppose I have $1 million in my account. I buy 100 shares of XYZ. Now, suppose XYZ goes up, and I have a profit on my 100 shares. I now want to own 200 shares, although the profit on my 100 shares is not yet $5,000 (i.e., XYZ has not yet gotten to $100). However, I buy another 100 shares anyhow. The schedule upon which I base my future buys (or sells) of XYZ (or any other stock while I own XYZ) is leverage—whether I borrow money to perform these transactions, or whether I use my own money. It is the schedule, the progressions, that constitutes leverage in this sense. If you understand this concept, you are well down the line toward understanding the new framework in asset allocation.

So, we see that *leverage* is a term that refers to either the degree to which we borrow money to take a position in an asset, or the *schedule* upon which we take further positions in assets (whether we borrow to do this or not).

That said, since the focus of the new framework is on *leverage*, we can easily see that it applies to speculative vehicles in the sense that leverage refers to the level of borrowing to take a position in a (speculative) asset. However, the new framework, in focusing on leverage, applies to all assets, including the most conservative, in the sense that leverage also refers to the progression, the schedule upon which we take (or remove) further positions in an asset. Ultimately, leverage in both senses is every bit as important as market timing. That is, the progression of asset accumulation and removal in even a very conservative bond fund is every bit as important as the bond market timing or the bond selection process.

Thus, the entire notion of *optimal f* not only applies to futures and option traders as well, but to any asset allocation scheme, and not just allocating among investment vehicles.

The trading world is vastly different today than just a few decades ago as a result of the proliferation of derivatives trading. Most frequently, a major characteristic with many derivatives is the leverage they bring to bear on an account. The old framework, the old two-dimensional E-V framework, was ill-equipped to handle problems of this sort. The modern environment *demands* a new asset allocation framework focused on the effects of leverage. The framework presented herein addresses exactly this.

This focus on leverage, more than any other explanation, is the main reason why the new framework is superior to its predecessors. Like the old framework, the new framework tells us optimal relative allocations among assets. But the new framework does far more. The new framework is dynamic—it tells us the immense consequences and payoffs of our schedule of taking (and removing) assets through time, giving us a *framework*, a map, of what consequences and rewards we can expect by following such-and-such a schedule. Certain points on the map may be more appealing than others to different individuals with different needs and desires. What may be optimal to one person may not be optimal to another. Yet this *map* allows us to see what we get and give up by progressing according to a certain schedule—something the earlier frameworks did not. This feature, this map of leverage space (and remember, leverage has two meanings here), distinguishes the new framework from its predecessors in many ways, and it alone makes the new framework superior.

Lastly, the new framework is superior to the old in that the user of the new framework can more readily see the consequences of his or her actions. Under the old framework, "So what if I have a little more V for a given E?" Under the new framework, you can see exactly what altitude that puts you at on the landscape, that is, exactly what multiple you make on your starting

stake (versus the peak of the landscape) for operating at different levels of leverage (remember, leverage has two meanings throughout this book), or exactly what kind of a *minimum* drawdown to expect for operating at different levels of *leverage*. Under the new framework, you can more readily see how important the asset allocation function is to your bottom line and your pain threshold.

To summarize, the new framework is superior to the older, two-dimensional, risk-competing-with-return frameworks primarily because the focus is on the *dynamics* of leverage. Secondarily, it is superior because the input is more straightforward, using scenarios (i.e., actual distributions that are "binned") unperverted by the misuse of the delusional correlation coefficient parameter, and because it will work on any distribution of returns. Lastly, users of the new framework will more readily be able to see the rewards and consequences of their actions.

MULTIPLE SIMULTANEOUS PLAYS

Refer to Figure 9.2 for our two-to-one coin-toss game. Now suppose you are going to play two of these very same games simultaneously. Each coin will be used in a separate game similar to the first game. Now what quantity should be bet? The answer depends upon the relationship of the two games. If the two games are not correlated to each other, then optimally you would bet 23% on each game (Figure 9.3). However, if there is perfect positive correlation, then you would bet 12.5% on each game. If you bet 25% or more

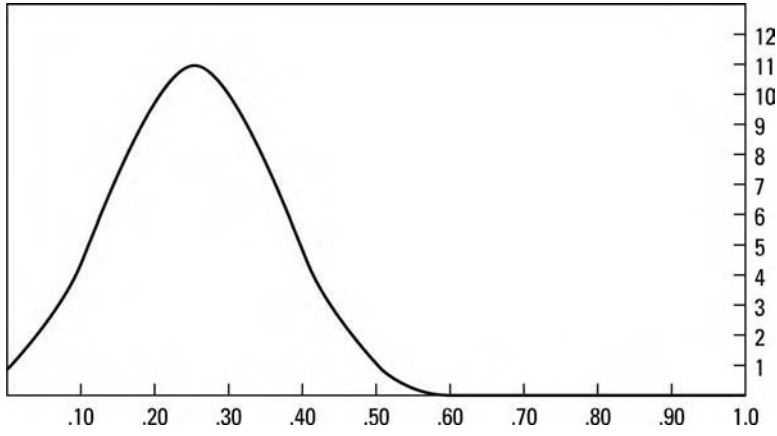

FIGURE 9.2 Two-to-one coin toss game, 40 plays. Ending multiple of starting stake betting different percentages of stake on each play

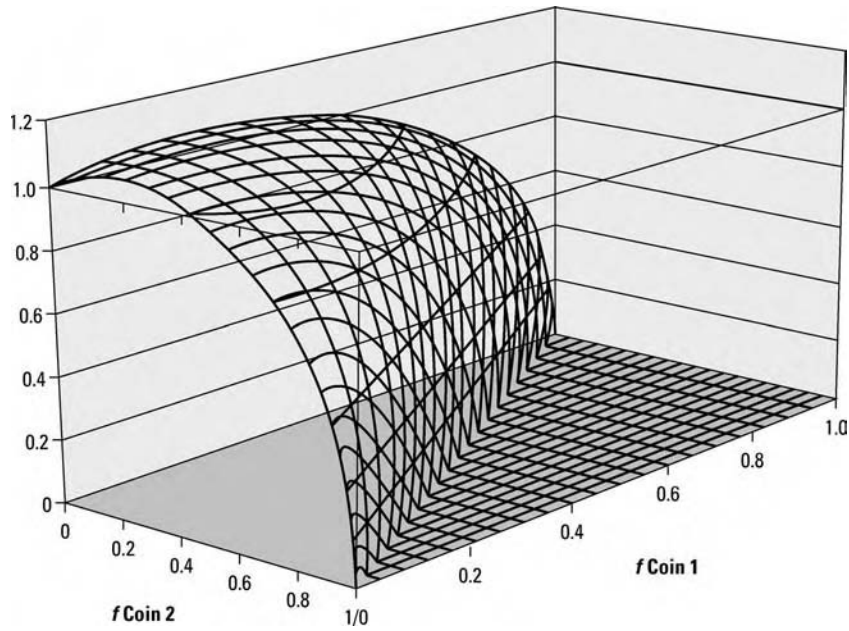

FIGURE 9.3 Two-to-one coin toss—one play

on each game, you will now go broke, with a probability that approaches certainty as the length of the game increases.

When you begin trading more than one market system, you no longer reside on a line that has a peak; instead, you reside in an $n + 1$ (where n = the number of market systems you are trading) dimensional terrain that has a single peak! In our single-coin-toss example, we had a peak on the line at 25%. Here we have one game ($n = 1$) and thus a two (i.e., $n + 1$) dimensional landscape (the line) with a single peak. When we play two of these games simultaneously, we now have a three-dimensional landscape (i.e., $n + 1$) within leverage space with a single peak. If the correlation coefficient between the coins is zero, then the peak is at 23% for the first game and 23% for the second as well. Notice that there is still only one peak, even though the dimensions of the landscape have increased!

When we are playing two games simultaneously, we are faced with a three-dimensional landscape, where we must find the highest point. If we were playing three games simultaneously, we would be looking for the peak in a four-dimensional landscape. The dimensions of the topography within which we must find a peak are equal to the number of games (markets and systems) we are playing plus one.

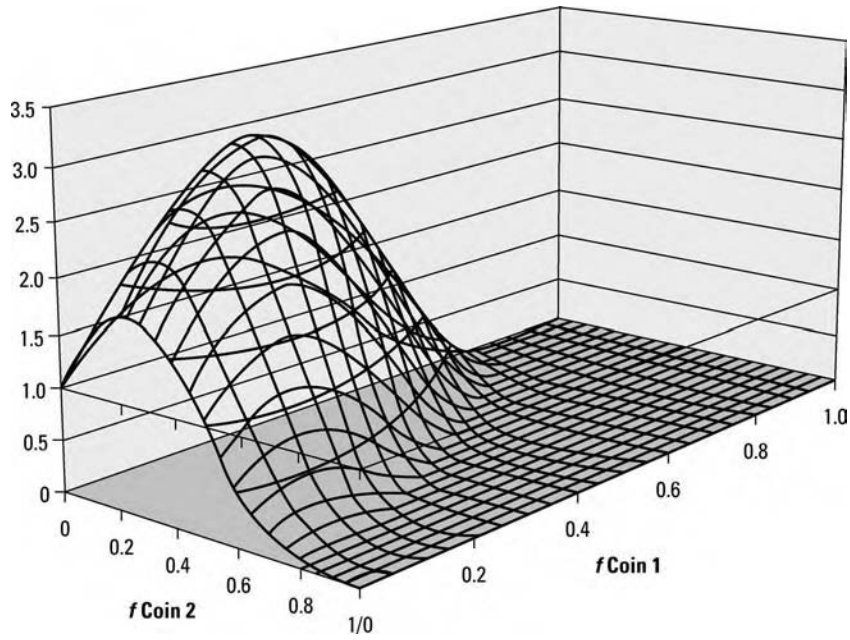

FIGURE 9.4 Two-to-one coin toss—10 plays

Notice, that as the number of plays increases, the peak gets higher and higher, and the difference between the peak and any other point on the landscape gets greater and greater (see Figures 9.3, 9.4, and 9.5). Thus, as more plays elapse, the difference between being at the peak and any other point increases. This is true regardless of how many markets or systems we are trading, even if we are trading only one.

To miss the peak is to pay a steep price. Recall in the simple single-coin-toss game the consequences of missing the peak. These consequences are no less when multiple simultaneous plays are involved. In fact, when you miss the peak in the $n + 1$-dimensional landscape, you will go broke faster than you would in the single game!

Whether or not we acknowledge these concepts, it does not affect the fact that *they are at work on us.* Remember, we can assign an f value to any trader in any market with any method at any time. If we are trading a single market system and we miss the peak of the f curve for that market system, we might, if we are lucky, make a fraction of the profits we should have made, while we will very likely endure greater drawdowns than we should have. If we are unlucky, we will go broke with certainty *even with an extremely profitable system!*

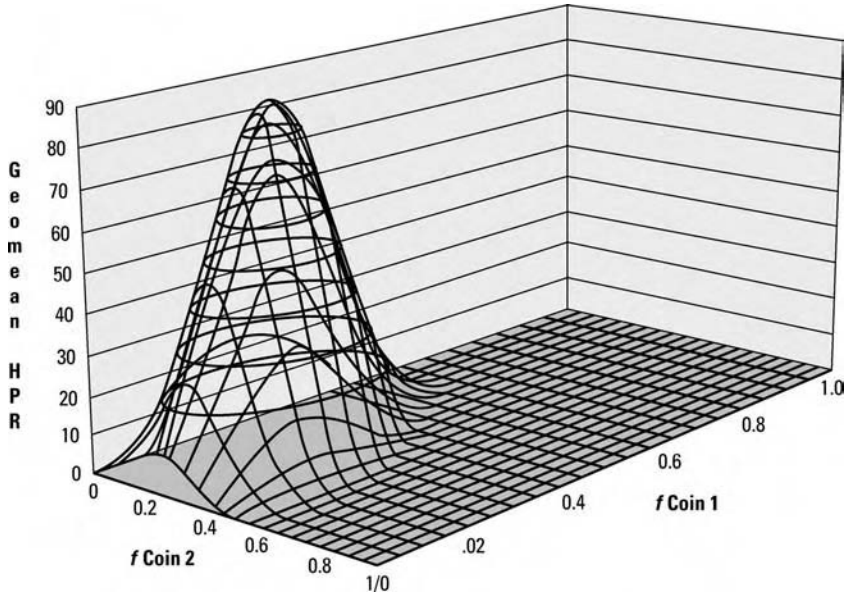

FIGURE 9.5 Two-to-one coin toss—40 plays

When we trade a portfolio of markets and/or systems, we simply magnify the effect of missing the peak of the curve in $n + 1$ space.

A COMPARISON TO THE OLD FRAMEWORKS

Let's take a look at a simple comparison of the results generated by this new framework versus those of the old E-V framework.

Suppose, for the sake of simplicity, we are going to play two simultaneous games. Each game will be the now-familiar two-to-one coin toss. Further assume that all of the pairwise correlations are zero. The new framework tells us that the optimal point, the peak in the three-dimensional $(n + 1)$ landscape is at 23% for both games.

The old framework, in addition to the zero values for the pairwise correlations, has .5 as the E value, the mean, and 2.25 as the V value, the variance. The result of this, through the old framework, generates .5 for both games.

This means that one-half of your account should be allocated toward each game. But what does this mean in terms of leverage? How much is a game? If a game is $1, the most I can lose, then .5 is way beyond the optimal of .23. How do I progress my stake as I go on? The correct answer, the

mathematically optimal answer with respect to leverage (including how I progress my stake as I go on), would be .5 of .46 of the account. But the old mean variance models do not tell me that. They are not attuned to the use of leverage (with both of its meanings). The answers tell me nothing of where I am in the $n + 1$ dimensional landscape. Also, there are important points within the $n + 1$ dimensional landscape other than the peak. For instance, as we will see in the next chapter, the points of inflection in the landscape are also very important. The old E-V models tell us nothing about any of this.

In fact, the old models simply tell us that allocating one-half of our stake to each of these games will be *optimal* in that you will get the greatest return for a given level of variance, or the lowest variance for a given level of return. How much you want to lever it is a matter of your utility—your personal preference.

In reality, though, there is an optimal point of leverage, an optimal place in the $n + 1$ dimensional landscape. There are also other important points in this landscape. When you trade, you automatically reside somewhere in this landscape (again, just because you do not acknowledge it does not mean it does not apply to you). The old models were oblivious to this. This new framework addresses this problem and has the users aware of the use/misuse of leverage within an optimal portfolio in a foremost sense. In short, the new framework simply yields more and more useful information than its predecessors.

Again, if a trader is utilizing two market systems simultaneously, then where he resides on the three-dimensional landscape is everything. Where he resides on it is every bit as important as his market systems, his timing, or his trading ability.

MATHEMATICAL OPTIMIZATION

Mathematical optimization is an exercise in finding a maximum or minimum value of an objective function for a given parameter(s). The objective function is, thus, something that can be solved only through an iterative procedure.

For example, the process of finding the optimal *f* for a single market system, or a single scenario spectrum, is an exercise in mathematical optimization. Here, the mathematical optimization technique can be something quite brutish like trying all *f* values from 0 to 1.0 by .01. The objective function can be one of the functions presented in Chapter 4 for finding the geometric mean HPR for a given value of *f* under different conditions. The parameter is that value for *f* being tried between 0 and 1.

The answer returned by the objective function, along with the parameters pumped into the objective function, gives us our coordinates at a certain point in $n + 1$ space. In the case of simply finding the optimal *f* for a single market system or a single scenario spectrum, n is 1, so we are getting coordinates in two-dimensional space. One of the coordinates is the *f* value sent to the objective function, and the other coordinate is the value returned by the objective function for the *f* value passed to it.

Since it is a little difficult for us to mentally picture any more than three dimensions, we will think in terms of a value of 2 for n (thus, we are dealing with the three-dimensional, i.e., $n + 1$, landscape). Since, for simplicity's sake, we are using a value of 2 for n, the objective function gives us the height or *altitude* in a three-dimensional landscape. We can think of the north-south coordinates as corresponding to the *f* value associated with one scenario spectrum, and the east-west coordinates as the *f* value associated with another scenario spectrum. Each scenario spectrum pertains to the possible outcomes for a given market system. Thus, we could say, for example, that the north-south coordinates pertain to the *f* value for such-and-such a market under such-and-such a system, and the east-west coordinates pertain to the *f* values of trading a different market and/or a different system, when both market systems are traded simultaneously.

The objective function gives us the altitude for a given set of *f* values. That is, the objective function gives us the altitude corresponding to a single east-west coordinate and a single north-south coordinate. That is, a single point where the length and depth are given by the *f* values we are pumping into the objective function, and the height at that point is the value returned by the objective function.

Once we have the coordinates for a single point (its length, depth, and height), we need a search procedure, a mathematical optimization technique, to alter the *f* values being pumped into the objective function in such a way so as to get us to the peak of the landscape as quickly and easily as possible.

What we are doing is trying to map out the terrain in the $n + 1$-dimensional landscape, because the coordinates corresponding to the peak in that landscape give us the optimal *f* values to use for each market system.

Many mathematical optimization techniques have been worked out over the years and many are quite elaborate and efficient. We have a number of these techniques to choose from. The burning question for us is, "Upon what objective function shall we apply these mathematical optimization techniques?" under this new framework. The objective function is the heart of this new framework in asset allocation, and we will discuss it and show examples of how to use it before looking at optimization techniques.

THE OBJECTIVE FUNCTION

The objective function we wish to maximize is the geometric mean HPR, simply called G:

$$G(f_1 \ldots f_n) = \left(\prod_{k=1}^{m} \text{HPR}_k\right)^{\left(1 \Big/ \sum_{k=1}^{m} \text{Prob}_k\right)} \tag{9.01}$$

where: n = The number of scenario spectrums (market systems or portfolio components).

m = The possible number of combinations of outcomes between the various scenario spectrums (market systems) based on how many scenarios are in each set. m = The number of scenarios in the first spectrum $*$ the number of scenarios in the second spectrum $* \ldots *$ the number of scenarios in the nth spectrum.

Prob = The sum of probabilities of all m of the HPRs for a given set of f values. Prob_k is the sum of the values in brackets $\{\}$ in Equation (9.02) for all m values of a given set of f values.

HPR = The holding period return of each k. This is given as:

$$\text{HPR}_k = \left(1 + \left(\sum_{i=1}^{n} (f_i * (-\text{PL}_{k,i}/\text{BL}_i))\right)\right)^{\text{Prob}_k} \tag{9.02}$$

where: n = The number of components (scenario spectrums, i.e., market systems) in the portfolio.

f_i = The f value being used for component i. f_i must be > 0, and can be infinitely high (i.e., can be greater than 1.0).

$\text{PL}_{k,i}$ = The outcome profit or loss for the ith component (i.e., scenario spectrum or market system) associated with the kth combination of scenarios.

BL_i = The worst outcome of scenario spectrum (market system) i.

We can estimate Prob_k in the earlier equation for G as:

$$\text{Prob}_k = \left(\prod_{i=1}^{n-1} \left(\prod_{j=i+1}^{n} P(i_k | j_k)\right)\right)^{(1/(n-l))} \tag{9.03}$$

The expression $P(i_k|j_k)$ is simply the joint probability of the scenario in the ith spectrum and the jth spectrum, corresponding to the kth combination of scenarios. For example, if we have three coins, each coin represents a scenario spectrum, represented by the variable n, and each spectrum contains two scenarios: heads and tails. Thus, there are eight $(2 * 2 * 2)$ possible combinations, represented by the variable m.

In Equation (9.01), the variable k proceeds from 1 to m, in *odometric* fashion:

Coin 1	**Coin 2**	**Coin 3**	***k***
t	t	t	1
t	t	h	2
t	h	t	3
t	h	h	4
h	t	t	5
h	t	h	6
h	h	t	7
h	h	h	8

That is, initially all spectrums are set to their worst (leftmost) values. Then, the rightmost scenario spectrum cycles through all of its values, after which the second rightmost scenario spectrum increments to the next (next right) scenario. You proceed as such again, with the rightmost scenario spectrum cycling through all of its scenarios, and when the second rightmost scenario spectrum has cycled through all of its values, the third rightmost scenario spectrum increments to its next scenario. The process is exactly the same as an odometer on a car, hence the term *odometrically*.

So in the expression $P(i_k|j_k)$, if k were at the value 3 above (i.e., $k = 3$), and i was 1 and j was 3, we would be looking for the joint probability of coin 1 (coming up tails and coin 3) coming up tails. Equation (9.03) helps us in estimating the joint probabilities of particular individual scenarios occurring in n spectrums simultaneously. To put it simply, if I have two scenario spectrums, at any given k I will have only one joint probability to incorporate. If I have three scenario spectrums, I will have three joint probabilities to incorporate (spectrums 1 and 2, spectrums 1 and 3, and spectrums 2 and 3). If four scenario spectrums, I will have six joint probabilities to compute using (9.03); if five scenario spectrums, then I have 10 joint probabilities to compute using (9.03). Quite simply, in (9.03) the number of joint probabilities you will have to incorporate at any $P(i)$ is:

$$n!/(n-2)!/2 = \text{number of joint probabilities required as input to (9.03)}$$

To demonstrate (9.03) in a simple manner, if I have three scenario spectrums (called A, B, and C), and each has two possible outcomes, H and T, then I want to find the multiplicative product of the probabilities of a given outcome of all three at each i, across all values of i (of which there are q).

So, if I have $n = 3$, then, at $k = 1$, I have the tails scenario (with a probability of .5) in all three scenario spectrums. Thus, to find the probability of this spectrum, I need to multiply the probability of $((A_T|B_T) \times (A_T|C_T) \times (B_T|C_T))\text{^}(1/(n– 1))$

$$= (.25 \times .25 \times .25))\ \text{^}\ (1/(3-1))$$
$$= .015625\ \text{^}\ (1/2) = .125$$

Note that this is a simple example. Our joint probabilities between any two scenarios from any of the three different scenario spectrums was always .25 in this case. In the real world, however, such conveniences are rare coincidences.

Equation (9.03) is merely an *estimate*, which makes a major assumption (that all elements move randomly with respect to all other elements, i.e. if we were to take a correlation coefficient of any pairwise elements, it would be zero). Note that we are constructing a $Prob_k$ here using (9.03); we are attempting to actually composite a joint probability of n events occurring simultaneously, knowing only the probabilities of pairwise occurrences of those events (at two scenario spectrums, this assumption is, in fact, *not* an assumption). In truth, this is an attempt to approximate the actual joint probability

For example, say I have three conditions called A, B, and C. A and B occur with .5 probability. A and C occur with .6 probability. B and C occur with .1 probability.

However, that probability of .1 of B and C's occurring may be 0 if A and B occur. It may be any value between 0 and 1 in fact. In order to determine then, what the probability of A, B, and C's occurring simultaneously is, I would have to look at when those three conditions actually *did* occur. I cannot infer the probability of all three events occurring simultaneously given the probabilities of their pairwise joint probabilities unless I am dealing with less than three elements or their pairwise correlations were all zero.

We need to derive the actual joint probability via empirical data, or accurately approximate the joint probabilities of occurrence among three or more simultaneous events. Equation (9.03) is invalid if there are more than two joint probabilities or the correlation coefficients between any pairwise elements is not 0. However, in the examples that follow in this chapter, we *will* use (9.03) merely as a proxy for whatever the actual joint probabilities

may be, for sake of illustration. We can create one complete objective function. Thus, we wish to maximize G as:

$$G(f_i \ldots f_n) = \left(\prod_{k=1}^{m} \left(\left(1 + \sum_{i=1}^{n} \left(f_i * \left(\frac{-\mathrm{PL}_{k,i}}{\mathrm{BL}_i} \right) \right) \right)^{\mathrm{Prob}_k} \right) \right)^{\left(1 / \sum_{k=1}^{m} \mathrm{Prob}_k \right)} \tag{9.04}$$

This is the objective function, the equation we wish to maximize. It is the equation or mathematical expression of this new framework in asset allocation. It gives you the *altitude*, the geometric mean HPR, in $n + 1$ space for the coordinates, the values of f used. It is *exact*, regardless of how many scenarios or scenario spectrums are used as input. *It is the objective function of the leverage space model.*

Although Equation (9.04) may look a little daunting, there isn't any reason to fear it. As you can see, Equation (9.04) is a lot easier to work with in the compressed form, expressed earlier in Equation (9.01).

Returning to our three coin example, suppose we win $2 on heads and lose $1 on tails. We have three scenario spectrums, three market systems, named Coin 1, Coin 2, and Coin 3. Two scenarios, heads and tails, comprise each coin, each scenario spectrum. We will assume, for the sake of simplicity, that the correlation coefficients of all three scenario spectrums (coins) to each other are zero.

We must therefore find three different f values. We are seeking an optimal f value for Coin 1, Coin 2, and Coin 3, as f_1, f_2, and f_3, respectively, that results in the greatest growth—that is, the combination of the three f values that results in the greatest geometric mean HPR [Equation (9.01) or (9.04)].

For the moment, we are not paying any attention to the optimization technique selected. The purpose here is to show how to perform the objective function. Since optimization techniques usually assign an initial value to the parameters, we will arbitrarily select .1 as the initial value for all three values of f.

We will use Equation (9.01) in lieu of (9.04) for the sake of simplicity. Equation (9.01) has us begin by cycling through all scenario set combinations, all values of k between 1 and m, compute the HPR of the scenario set combination per Equation (9.02), and multiply all of these HPRs together. When we perform Equation (9.02) each time, we must keep track of the $\mathrm{Prob_k}$ values, because we will need the sum of these values later.

Thus, we start at $k = 1$, where scenario spectrum 1 (Coin 1) is tails, as are the other two scenario spectrums (coins).

We can rewrite Equation (9.02) as:

$$\begin{aligned}\text{HPR}_k &= (1+C)^x \\ C &= \sum_{i=1}^{n}(f_i * (-\text{PL}_{k,i}/\text{BL}_i)) \\ x &= \left(\prod_{k=1}^{n-1}\left(\prod_{j=i+1}^{n} P(i_k|j_k)\right)\right)^{(1/(n-1))}\end{aligned}$$

Notice that the exponent in Equation (9.02), which we must keep track of, is expressed as the variable x in Equation (9.02a). This is also expressed in Equation (9.03).

So, to obtain C, we simply go through each scenario spectrum, taking the outcome of the scenario currently being used in that spectrum as dictated by k, dividing its negative by the scenario in that spectrum with the worst outcome, and multiplying this quotient by the f value being used with that scenario spectrum. As we go through all of the scenario spectrums, we total these values.

The variable i is the scenario spectrum we are looking at. The biggest loss in scenario spectrum 1 is tails, which sees a loss of one dollar (i.e., -1). Thus, BL_1 is -1 (as will be BL_2 and BL_3 since the biggest loss in each of the other two scenario spectrums—the other two coins—is -1). The associated PL, that is, the outcome of the scenario in spectrum i corresponding to the scenario in that spectrum that k points to, is -1 in scenario spectrum 1 (as it is in the other two spectrums). The f value is currently .1 (as it also is now in the other two spectrums). Thus:

$$\begin{aligned}C &= \sum_{i=1}^{n}\left(f_i * \left(\frac{-\text{PL}_{k,i}}{\text{BL}_i}\right)\right) \\ C &= \left(.1 * \left(\frac{--1}{-1}\right)\right) + \left(.1 * \left(\frac{--1}{-1}\right)\right) + \left(-1 * \left(\frac{--1}{-1}\right)\right) \\ C &= (.1 * -1) + (.1 * -1) + (.1 * -1) \\ C &= -.1 + -.1 + -.1 = -.3\end{aligned}$$

Notice that the PLs are negative and, since PL has a minus sign in front of it, that makes them positive.

Now we take the value for C in Equation (9.02) above and add 1 to it, obtaining .7 (since $1 + -.3 = .7$). Now we must figure the exponent, the variable x in Equation (9.02) above.

$P(i_k|j_k)$ means, simply, the joint probability of the scenario in spectrum i pointed to by k, and the scenario in spectrum j pointed to by k. Since k is presently 1, it points to tails in all three scenario spectrums. To find x, we

simply take the sum of the joint probabilities of the scenarios in spectrum 1 and 2 times the joint probability of the scenarios in spectrum 1 and 3, times the joint probabilities of the scenarios in spectrums 2 and 3. Expressed differently:

i	*j*
1	2
1	3
2	3

If there were four spectrums, we would take the product of all the joint probabilities as:

i	*j*
1	2
1	3
1	4
2	3
2	4
3	4

Since all of our joint probabilities are .25, we get for x:

$$x = \left(\prod_{i=1}^{n-1}\left(\prod_{j=i+1}^{n} P(i_k|j_k)\right)\right)^{(1/(n-1))}$$

$$x = (.25 * .25)^{(1/(n-1))}$$

$$x = (.015625)^{1/(3-1)}$$

$$x = (.015625)^{1/2}$$

$$x = .125$$

Thus, x equals .125, which represents the joint probability of the kth combination of scenarios. (Note that we are going to determine a joint probability of three random variables by using joint probabilities of two random variables!)

Thus, $\text{HPR}_k = .7^{.125} = .9563949076$ when $k = 1$. Per Equation (9.02), we must figure this for all values of k from 1 through m (in this case, m equals 8). Doing this, we obtain:

k	HPR_k	Prob_k
1	0.956395	0.125
2	1	0.125
3	1	0.125
4	1.033339	0.125
5	1	0.125
6	1.033339	0.125
7	1.033339	0.125
8	1.060511	0.125

Summing up all the Prob_k, given by Equation (9.03), per Equation (9.04), we get 1. Now, taking the product of all of the HPRs, per Equations (9.01) and (9.04), we obtain 1.119131. Performing Equation (9.01), then, we get a value of G of 1.119131 which corresponds to the f values .1, .1, .1 for f_1, f_2, and f_3, respectively.

$$G(.1, .1, .1) = \left(\prod_{k=1}^{m} \text{HPR}_k\right)^{\left(1 \Big/ \sum_{k=1}^{m} \text{Prob}_k\right)}$$

$$G(.1, .1, .1) = (.956395 * 1 * .1 * 1.033339 * 1 * 1.033339 * 1.033339 * 1.033339 * 1.0605011)^{(1/(.125+.125+.125+.125+.125+.125+.125+.125))}$$

$$G(.1, .1, .1) = (1.119131)^{(1/1)}$$

$$G(.1, .1, .1) = 1.119131$$

Now, depending upon what mathematical optimization method we were using, we would alter our f values. Eventually, we would find our optimal f values at .21, .21, .21 for f_1, f_2, and f_3, respectively. This would give us:

k	HPR_k	Prob_k
1	0.883131	0.125
2	1	0.125
3	1	0.125
4	1.062976	0.125
5	1	0.125
6	1.062976	0.125
7	1.062976	0.125
8	1.107296	0.125

Thus, Equation (9.01) gives us:

$$G(.21, .21, .21) = \left(\prod_{k=1}^{m} \text{HPR}_k\right)^{\left(1 \Big/ \sum_{k=1}^{m} \text{Prob}_k\right)}$$

$$G(.21, .21, .21) = (.883131 * 1 * .1 * 1.062976 * 1 * 1.062976 * 1.062976 * 1.107296)^{(1/(.125+.125+.125+.125+.125+.125+.125+.125))}$$

$$G(.21, .21, .21) = 1.174516^{(1/1)}$$

$$G(.21, .21, .21) = 1.174516$$

This is the *f* value combination that results in the greatest *G* for these scenario spectrums. Since this is a very simplified case, that is, all scenario spectrums were identical, and all had correlation of zero between them, we ended up with the same *f* value for all three scenario spectrums of .21. Usually, this will not be the case, and you will have a different *f* value for each scenario spectrum.

Now that we know the optimal *f* values for each scenario spectrum, we can determine how much those decimal *f* values are, in currency, by dividing the largest loss scenario in each of the spectrums by the negative optimal *f* for each of those spectrums. For example, for the first scenario spectrum, Coin 1, we had a largest loss of −1. Dividing −1 by the negative optimal *f*, −.21, we obtain 4.761904762 as *f*$ for Coin 1.

To summarize the procedure, then:

1. Start with an *f* value set for $f_1 \ldots f_n$ where n is the number of components in the portfolio, that is, market systems or scenario spectrums. This initial *f* value set is given by the optimization technique selected.
2. Go through the combinations of scenario sets k from 1 to m, odometrically, and calculate an HPR for each k, multiplying them all together. While doing so, keep a running sum of the exponents of the HPRs.
3. When k equals m, and you have computed the last HPR, the final product must be taken to the power of 1, divided by the sum of the exponents (probabilities) of all the HPRs, to get *G*, the geometric mean HPR.
4. This geometric mean HPR gives us one *altitude* in $n + 1$ space. We wish to find the peak in this space, so we must now select a new set of *f* values to test to help us find the peak. This is the mathematical optimization process.

MATHEMATICAL OPTIMIZATION VERSUS ROOT FINDING

Equations have a left and a right side. Subtracting the two makes the equation equal to 0. In *root finding*, you want to know what values of the

independent variable(s) make the answer of this equation equal to 0 (these are the *roots*). There are traditional root-finding techniques, such as the *Newton-Rapheson* method, to do this.

It would seem that root finding is related to mathematical optimization in that the first derivative of an optimized function (i.e., extremum located) will equal 0. Thus, you would assume that traditional root-finding techniques, such as the Newton-Rapheson method, could be used to solve optimization problems (careful to use what is regarded as an optimization technique to solve for the roots of an equation can lead to a Pandora's box of problems).

However, our discussion will concern only optimization techniques and not root finding techniques per se. The single best source for a listing of these techniques is *Numerical Recipes* and much of the following section on optimization techniques is referenced therefrom.[3]

OPTIMIZATION TECHNIQUES

Mathematical optimization, in short, can be described as follows: You have a function (we call it G), the objective function, which depends on one or more independent variables (which we call $f_1 \ldots f_n$). You want to find the value(s) of the independent variable(s) that results in a minimum (or sometimes, as in our case, a maximum) of the objective function. Maximization or minimization is essentially the same thing (that is one person's G is another person's $-G$).

In the crudest case, you can optimize as follows: Take every combination of parameters, run them through the objective function, and see which produce the best results. For example, suppose we want to find the optimal f for two coins tossed simultaneously, and we want the answer to be precise to .01. We could, therefore, test Coin 1 at the 0.0 level, while testing Coin 2 at the 0.01 level, then .01, .02, and proceed until we have tested Coin 2 at the 1.0 level. Then, we could go back and test with Coin 1 at the .01 level, and cycle Coin 2 through all of its possible values while holding Coin 1 at the .01 level. We proceed until both levels are at their maximum, that is, both values equal 1.0. Since each variable in this case has 101 possible values (0 through 1.0 by .01 inclusive), there are $101 * 101$ combinations which must be tried, or 10,201 times the objective function must be evaluated.

We could, if we wanted, demand precision greater than .01. Suppose we wanted precision to the .001 level. Then we would have $1{,}001 * 1{,}001$

[3]William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling, *Numerical Recipes: The Art of Scientific Computing*, New York: Cambridge University Press, 1986.

combinations that we would need to try, or 1,002,001 times the objective function would have to be calculated. If we were then to include three variables rather than just two, and demand .001 precision this way, we would then have to evaluate the objective function 1001 * 1001 * 1001, or 1,003,003,001; that is, we would have to evaluate the objective function in excess of one billion times. We are using only three variables and we are demanding precision to only .001!

Although this crude case of optimizing has the advantage of being the most robust of all optimization techniques, it is also has the dubious distinction of being too slow to apply to most problems.

Why not cycle through all variables for the first variable and get its optimal; then cycle through all variables for the second while holding the first at its optimal; get the second variable's optimal, so that you now have the optimal for the first two parameters; go find the optimal for the third while setting the first two to their optimal, and so on, until you have solved the problem?

The problem with this second approach is that it is often impossible to find the optimum parameter set this way. Notice that by the time we get to the third variable, the first two variables equal their optimum as if there were no other variables. Thus, when the third variable is optimized, with the first two variables set to their optimums, they interfere with the solution of the third optimum. What you would end up with is not the optimum parameter set of the three variables, but, rather, an optimum value for the first parameter, an optimum for the second when the first is set to its optimum, an optimum for the third when the first is set to its optimum, and the second set to a suboptimum, but optimum given the interference of the first, and so on. It may be possible to keep cycling through the variables and eventually resolve to the optimum parameter set, but with more than three variables, it becomes more and more lengthy, if at all possible, given the interference of the other variables.

There exist superior techniques that have been devised, rather than the two crude methods described, for mathematical optimization. This is a fascinating branch of modern mathematics, and I strongly urge you to study it, simply in the hope that you derive a fraction of the satisfaction from the study as I have.

An extremum, that is the maximum or minimum, can be either *global* (truly the highest or lowest value) or *local* (the highest or lowest value in the immediate neighborhood). To truly know a global extremum is nearly impossible, since you do not know the range of values of the independent variables. If you do not know the range, then you have simply found a local extremum. Therefore, oftentimes, when people speak of a global extremum, they are really referring to a local extremum over a very wide range of values for the independent variables.

There are a number of techniques for finding the maximum or minimum in such cases. Usually, in any type of mathematical optimization, there are *constraints* placed on the variables, which must be met with respect to the extremum. For example, in our case, there are the constraints that all independent variables (the *f* values) must be greater than or equal to zero. Oftentimes, there are constraining functions that must be met [i.e., other functions involving the variable(s) used which must be above/below or equal to certain values]. *Linear programming*, including the *simplex algorithm*, is one very well developed area of this type of constrained optimization, but will work only where the function to be optimized and the constraint functions are linear functions (first-degree polynomials).

Generally, the different methods for mathematical optimization can be broken down by the following categories, and the appropriate technique selected:

1. Single-variable (two-dimensional) vs. multivariable (three- or more dimensional) objective functions.
2. Linear methods vs. nonlinear methods. That is, as previously mentioned, if the function to be optimized and the constraint functions are linear functions (i.e., do not have exponents greater than one to any of the terms in the functions), there are a number of very well developed techniques for solving for extrema.
3. Derivatives. Some methods require computation of the first derivative of the objective function. In the multivariable case, the first derivative is a vector quantity called the *gradient*.
4. Computational efficiency. That is, you want to find the extremum as quickly (i.e., with as few computations) and easily (something to consider with those techniques which require calculation of the derivative) as possible, using as little computer storage as possible.
5. Robustness. Remember, you want to find the extremum that is local to a very wide range of parameter values, to act as a surrogate global extremum. Therefore, if there is more than one extremum in this range, you do not want to get hung up on the less extreme extremum.

In our discussion, we are concerned only with the multidimensional case. That is, we concern ourselves only with those optimization algorithms that pertain to two or more variables (i.e., more than one scenario set). *In searching for a single* f *value, that is, in finding the* f *of one market system or one scenario set, parabolic interpolation, as detailed in Chapter 4*, Portfolio Management Formulas, *will generally be the quickest and most efficient technique.*

In the multidimensional case, there are many good algorithms, yet there is no perfect algorithm. Some methods work better than others for certain types of problems. Generally, personal preference is the main determinant in selecting a multidimensional optimization technique (provided one has the computer hardware necessary for the chosen technique).

Multidimensional techniques can be classified according to five broad categories.

First are the *hill-climbing simplex methods*. These are perhaps the least efficient of all, if the computational burden gets a little heavy. However, they are often easy to implement and do not require the calculation of partial first derivatives. Unfortunately, they tend to be slow and their storage requirements are on the order of $n^{2.}$

The second family are the *direction set methods*, also known as the *line minimization methods* or *conjugate direction methods*. Most notable among these are the various methods of Powell. These are more efficient, in terms of speed, than the hill-climbing simplex methods (not to be confused with the simplex algorithm for linear functions mentioned earlier), do not require the calculation of partial first derivatives, yet the storage requirements are still on the order of n^2.

The third family is the *conjugate gradient methods*. Notable among these are the Fletcher-Reeves method and the closely related Polak-Ribiere method. These tend to be among the most efficient of all methods in terms of speed and storage (requiring storage on the order of n times x), yet they do require calculations of partial first derivatives.

The fourth family of multidimensional optimization techniques are the *quasi-Newton*, or *variable metric methods*. These include the Davidson-Fletcher-Powell (DFP) and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithms. Like the conjugate gradient methods, these require calculation of partial first derivatives, tend to rapidly converge to an extremum, yet these require greater storage, on the order of n^2. However, the tradeoff to the conjugate gradient methods is that these have been around longer, are in more widespread use, and have greater documentation.

The fifth family is the *natural simulation* family of multidimensional optimization techniques. These are by far the most fascinating, as they seek extrema by simulating processes found in nature, where nature herself is thought to seek extrema. Among these techniques are the *genetic algorithm* method, which seeks extrema through a survival-of-the-fittest process, and *simulated annealing*, a technique which simulates crystallization, a process whereby a system finds its minimum energy state. These techniques tend to be the most robust of all methods, nearly immune to local extrema, and can solve problems of gigantic complexity. However, they are not necessarily the quickest, and, in most cases, will not be. These techniques are still so new that very little is known about them yet.

Although you can use any of the aforementioned multidimensional optimization algorithms, I have opted for the genetic algorithm because it is perhaps the single most robust mathematical optimization technique, aside from the very crude technique of attempting every variable combination.

It is a *general* optimization and search method that has been applied to many problems. Often it is used in neural networks, since it has the characteristic of scaling well to noisy or large nonlinear problems. Since the technique does not require gradient information, it can also be applied to discontinuous functions, as well as empirical functions, just as it is applied to analytic functions.

The algorithm, although frequently used in neural networks, is not limited solely to them. Here, we can use it as a technique for finding the optimal point in the $n + 1$ dimensional landscape.

THE GENETIC ALGORITHM

In a nutshell, the algorithm works by examining many possible candidate solutions and ranking them on how well their value output, by whatever objective function, is used. Then, like the theory of natural selection, the most fit survive and reproduce a new generation of candidate solutions, which inherit characteristics of both *parent* solutions of the earlier generation. The average fitness of the population will increase over many generations and approach an optimum.

The main drawback to the algorithm is the large amount of processing overhead required to evaluate and maintain the candidate solutions. However, due to its robust nature and effective implementation to the gamut of optimization problems, however large, nonlinear, or noisy, it is this author's contention that it will become the de facto optimization technique of choice in the future (excepting the emergence of a better algorithm which possesses these desirable characteristics). As computers become ever more powerful and inexpensive, the processing overhead required of the genetic algorithm becomes less of a concern. Truly, if processing speed were zero, if speed were not a factor, the genetic algorithm would be the optimization method of choice for nearly all mathematical optimization problems.

The basic steps involved in the algorithm are as follows:

1. *Gene length.* You must determine the length of a *gene.* A gene is the binary representation of one member of the population of candidate solutions, and each member of this population carries a value for each variable (i.e., an f value for each scenario spectrum). Thus, if we allow a gene length of 12 times the number of scenario spectrums, we have 12 bits

assigned to each variable (i.e., *f* value). Twelve bits allows for values in the range of 0 to 4095. This is figured as:

$$2^0 + 2^1 + 2^2 + \ldots + 2^{11} = 4095$$

Simply take 2 to the 0th power plus 2 to the next power, until you reach the power of the number of bits minus 1 (i.e., 11 in this case). If there are, say, three scenario spectrums, and we are using a length of 12 bits per scenario spectrum, then the length of a gene for each candidate solution is $12 * 3 = 36$ bits. That is, the gene in this case is a string of 36 bits of 1s and 0s.

Notice that this method of encoding the bit strings only allows for integer values. We can have it allow for floating-point values as well by using a uniform divisor. Thus, if we select a uniform divisor of, say, 1,000, then we can store values of 0/1000 to 4095/1000, or 0 to 4.095, and get precision down to .001.

What we need then is a routine to convert the candidate solutions to encoded binary strings and back again.

2. *Initialization.* A starting population is required—that is, a population of candidate solutions. The bit strings of this first generation are encoded randomly. Larger population sizes make it more likely that we will find a good solution, but they require more processing time.

3. *Objective function evaluation.* The bit strings are decoded to their decimal equivalents, and are used to evaluate the objective function. (The objective function, for example, if we are looking at two scenario spectrums, gives us the *Z* coordinate value, the altitude of the three-dimensional terrain, assuming the *f* values of the respective scenario spectrums are the *X* and *Y* coordinates.) This is performed for all candidate solutions, and their objective functions are saved. (*Important:* Objective function values must be non-negative!)

4. *Reproduction*

- **a.** *Scaling based upon fitness.* The objective functions are now scaled. This is accomplished by first determining the lowest objective function of all the candidate solutions, and subtracting this value from all candidate solutions. The results of this are summed up. Then, each objective function has the smallest objective function subtracted from it, and the result is divided by the sum of these, to obtain a fitness score between 0 and 1. The sums of the fitness scores of all candidate solutions will then be 1.0.

- **b.** *Random selection based upon fitness.* The scaled objective functions are now aligned as follows. If, say, the first objective function

has a scaled fitness score of .05, the second has one of .1, and the third .08, then they are set up in a selection scheme as follows:

First candidate	0 to .05
Second candidate	.05 to .15
Third candidate	.15 to .23

This continues until the last candidate has its upper limit at 1.0.

Now, two random numbers are generated between 0 and 1, with the random numbers determining from the preceding selection scheme who the two parents will be. Two parents must now be selected for each candidate solution of the next generation.

c. *Crossover*. Go through each bit of the *child*, the new population candidate. Start by copying the first bit of the first parent to the first bit of the child. At each bit carryover, you must also generate a random number. If the random number is less than or equal to (probability of crossover/gene length), then switch to copying the bits over from the other parent. Thus, if we have three scenario spectrums and 12 bits per each variable, then the gene length is 36. If we use a probability of crossover of .6, then the random number generated at any bit must be less than .6/36, or less than .01667, in order to switch to copying the other parent's code for subsequent bits. Continue until all the bits are copied to the child. This must be performed for all new population candidates.

Typically, probabilities of crossover are in the range .6 to .9. Thus, a .9 probability of crossover means there is a 90% chance, on average, that there will be crossover to the child, that is, a 10% chance the child will be an exact replicant of one of the parents.

d. *Mutation*. While copying over each bit from parent to child, generate a second random number. If this random number is less than or equal to the probability of mutation, then toggle that bit. Thus, a bit which is 0 in the parent becomes 1 in the child and vice versa. Mutation helps maintain diversity in the population. The probability of mutation should generally be some small value (i.e., $<=.001$); otherwise the algorithm tends to deteriorate into a random search. As the algorithm approaches an optimum, however, mutation becomes more and more important, since crossover cannot maintain genetic diversity in such a localized space in the $n + 1$ terrain.

Now you can go back to step three and perform the process for the next generation. Along the way, you must keep track of the highest objective function returned and its corresponding gene. Keep repeating the process

until you have reached X unimproved generations, that is, X generations where the best objective function value has not been exceeded. You then quit, at that point, and use the gene corresponding to that best objective function value as your solution set.

For an example of implementing the genetic algorithm, suppose our objective function is one of the form:

$$Y = 1500 - (X - 15)^2$$

For the sake of simplicity in illustration, we will have only a single variable; thus, each population member carries only the binary code for that one variable.

Upon inspection, we can see that the optimal value for X is 15, which would result in a Y value of 1500. However, rarely will we know what the optimal values for the variables are, but for the sake of this simple illustration, it will help if we know the optimal so that we can see how the algorithm takes us there.

Assume a starting population of three members, each with the variable values encoded in five-bit strings, and each initially random:

First Generation				
Individual #	***X***	**Binary *X***	***Y***	**Fitness Score**
1	10	01010	1475	.4751
2	0	00000	1275	0
3	13	01101	1496	.5249

Now, through random selection based on fitness, Individual 1 for the second generation draws Parents 1 and 3 from the first generation (note that Parent 2, with a fitness of 0, has died and will not pass on its genetic characteristics). Assume that random crossover occurs after the fourth bit, so that Individual 1 in the second generation inherits the first four bits from Individual 1 of the first generation, and the last bit from Individual 3 of the first generation, producing 01011 for Individual 1 of the second generation.

Assume Individual 2 for the second generation also draws the same parents; crossover occurs only after the first and third bits. Thus, it inherits bit 0 from Individual 1 in the first generation, bit 11 as the second and third bits from the third individual in the first generation, and the last two bits from the first individual of the first generation, producing 01110 as the genetic code for the second individual in the second generation.

Now, assume that the third individual of the second generation draws Individual 1 as its first parent as well as its second. Thus, the third individual

in the second generation ends up with exactly the same genetic material as the first individual in the first generation, or 01010.

Second Generation		
Individual #	***X***	**Binary *X***
1	11	01011
2	14	01110
3	10	01010

Now, through random mutation, the third bit of the first individual is flipped, and the resulting values are used to evaluate the objective function:

Second Generation				
Individual #	***X***	**Binary *X***	***Y***	**Fitness Score**
1	15	01111	1500	.5102
2	14	01110	1499	.4898
3	10	01010	1475	0

Notice how the average *Y* score has gone up, or evolved, after two generations.

IMPORTANT NOTES

It is often advantageous to carry the strongest individual's code to the next generation in its entirety. By so doing, good solution sets are certain to be maintained, and this has the effect of expediting the algorithm. Then, you can work to aggressively maintain genetic diversity by increasing the values used for the probability of crossover and the probability of mutation. I have found that you can work with a probability of crossover of 2, a probability of mutation of .05, and converge to solutions quicker, provided you retain the code of the most fit individual from one generation to the next, which keeps the algorithm from deteriorating to a random search.

As population size approaches infinity, that is, as you use a larger and larger value for the population size, the answer converged upon is exact. Likewise, with the unimproved generations parameter, as it approaches

infinity—that is, as you use a larger and larger value for unimproved generations—the answer converged upon is exact. However, both of these parameter increases are at the expense of extra computing time.

The algorithm can be time intensive. As the number of scenario sets increases, and the number of scenarios increases, the processing time grows geometrically. Depending upon your time constraints, you may wish to keep your scenario sets and the quantity of scenarios to a manageable number. The genetic algorithm is particularly appropriate as we shall see by Chapter 12, where we find the landscape of leverage space to be discontinuous for our purposes.

Once you have found the optimal portfolio, that is, once you have *f* values, you simply divide those *f* values by the largest loss scenario of the respective scenario spectrums to determine the *f* $ for that particular scenario spectrum. This is exactly as we did in the previous chapter for determining how many contracts to trade in an optimal portfolio.