As the Markov chain progresses, we keep track of the minimal route length observed so far and the route that produced it.

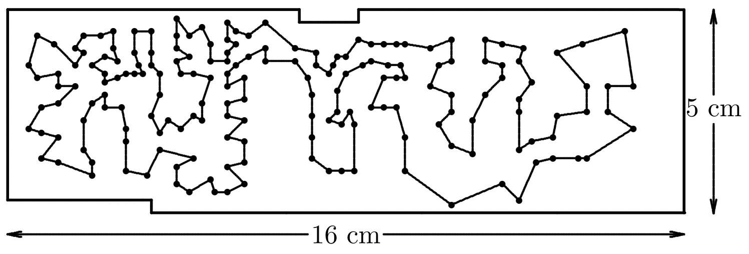
As with previous applications, a virtue of our notion of neighbor is that the computation of $\Delta E(x, y)$ is easy and does not grow in computational time as the number of sites K increases. The dashed parts of the route in the previous figure are unchanged (although the sites from c_i to c_j are toured in reverse order). Hence we have

$$\Delta E(x,y) \;=\; \big[d(c_{i-1},c_j)+d(c_i,c_{j+1})\big]-\big[d(c_{i-1},c_i)+d(c_j,c_{j+1})\big].$$

Naturally one could compute $E(x)$ and $E(y)$ separately using (1) and then compute the change in energy as $\Delta E(x, y) = E(y) - E(x)$, but this would be much more time consuming.

This is implemented in `RouteOptimization.cpp`. This algorithm produces good results when $T = 0.07$. At that temperature, the best route found occurred at about 51 million steps into the Markov chain and produced a route length of just 84.916 centimeters. This route may not be optimal, but it is probably pretty close.

Route length = 84.916 cm, found at step 51,323,843



For more on TSP go to `www.math.uwaterloo.ca/tsp`. See also [Ri].

§18. Portfolio Optimization

The file `V.txt` contains the 50×50 positive definite sample covariance matrix $\mathbf{V} = (v_{ij})$ for the month-to-month change in value ΔV per 1 dollar investment for each of 50 stocks from the S&P 100 over a five year time period. In what follows, a

portfolio, call it $\mathbf{x} = (x_1, \ldots, x_{50})^T$, is a 50×1 column vector whose components sum to \$100. The variance of the percent monthly return of such a portfolio is

$$\mathrm{Var}\,(\mathbf{x}) \;=\; \mathbf{x}^T\mathbf{V}\mathbf{x} \;=\; \sum_{i=1}^{50}\sum_{j=1}^{50} x_i v_{ij} x_j.$$

We will use Metropolis to determine minimum variance portfolios subject to certain types of constraints.

No Constraints. Initially we impose no constraints on the portfolio. Both long and short positions are allowed. The only requirement is that the components sum to 100. Simple quadratic programming addresses this problem. In particular, the minimum variance portfolio is $\mathbf{x}^* = 100 \cdot \frac{1}{c} \cdot \mathbf{V}^{-1}\mathbf{e}$, where $\mathbf{e}_{50\times 1} = (1, 1, \ldots, 1)^T$ and $c = \mathbf{e}^T\mathbf{V}^{-1}\mathbf{e}$ (see [LX], e.g.).

The set of all portfolios has a continuum of values, so to employ Metropolis we must discretize the state space. With $\epsilon = 0.001$, we'll take

$$S \;=\; \big\{\mathbf{x} = (x_1, \ldots, x_{50})^T : \text{each } x_i \in \epsilon \times \mathbb{Z} \text{ and } \sum_i x_i = 100\big\},$$

so, for example, 10.552 is a legitimate portfolio component but 10.5526 is not. This state space, while discrete, is now countably infinite. That will not be a problem, however, as $\mathrm{Var}\,(\mathbf{x})$ grows like $||\mathbf{x}||^2$ and the Markov chain will not wander outside of a bounded region of S. As for the notion of neighbor, we'll say that $\mathbf{x} \leftrightarrow \mathbf{y}$ if, for some two components $i \neq j$ we have $y_i = x_i + \epsilon$ and $y_j = x_j - \epsilon$, with $x_k = y_k$ for $k \neq i$ or j. The Markov chain will alter portfolio composition bit-by-bit. This notion is easily seen to comply with the good neighbor rules — in particular, each state has $50 \times 49 = 2{,}450$ neighbors. We will take $E(\mathbf{x}) = \mathrm{Var}\,(\mathbf{x})$, the quantity we seek to minimize.

About the Temperature Parameter. Suppose $\mathbf{x} \neq \mathbf{y}$ are any two distinct portfolios — they are permitted to have both long and short positions. Then $\mathbf{p} = \lambda\mathbf{x} + (1-\lambda)\mathbf{y}$ is another portfolio for any real number λ. A little algebra shows that $\mathrm{Var}\,(\mathbf{p})$ is quadratic in λ — of the form $a\lambda^2 + b\lambda + c$ for some real numbers b, and c and with $a = \mathrm{Var}\,(\mathbf{x} - \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T\mathbf{V}(\mathbf{x} - \mathbf{y})$. Since

$\mathbf{x} - \mathbf{y} \neq \mathbf{0}$ and $\mathbf{V}$ is positive definite we see that $a > 0$. Hence $\text{Var}(\mathbf{p})$ is minimized for a *unique* value of λ and $\mathbf{x}$ and $\mathbf{y}$ cannot both be local minima for $\text{Var}(\cdot)$. It follows that there are no stable states in S (other than the global minimum) and we may safely use the zero-temperature dynamics: we'll take $T = 0$. In this application we don't know the minimal variance in advance. However, we do know that the global minimum is the only stable state. We'll run the Markov chain testing periodically if we're at a stable state and stop when we find one.

This is implemented in `NoConstraints.cpp`. The resulting Metropolis portfolio (variance = 0.45442) is shown below. The true optimal $\mathbf{x}^*$ (also with variance = 0.45442) agrees at every component to within a few pennies.

Portfolio composition with no constraints

MS	14.72	PCLN	-12.83
RTN	24.70	HD	-7.86
OXY	16.82	MDT	7.57
TWX	-2.19	PFE	-35.93
F	16.24	CAT	-16.75
AIG	-0.71	JNJ	-51.19
ACN	-22.50	GOOGL	-12.11
HON	8.59	CELG	-2.92
QCOM	26.23	NEE	30.56
BA	-17.87	DD	-0.16
UNH	-4.98	WFC	-0.38
PG	-2.65	KHC	1.18
HAL	20.57	CVX	4.53
INTC	-8.49	MMM	9.69
BLK	-13.11	MSFT	0.22
UPS	4.14	FDX	13.52
UNP	-9.20	VZ	24.93
V	17.29	KO	12.36
NKE	8.24	TGT	24.58
BIIB	6.31	WBA	-14.64
SLB	-16.70	MCD	15.07
COST	-4.10	CVS	-13.63
SPG	12.96	KMI	-4.25
MO	6.49	LLY	31.18
AMZN	8.10	GD	8.38

No Short Positions. Here we insist that each component x_i of the solution is non-negative. We implement this not by restricting the state space S but rather by modifying the energy function. Specifically, we take

$$E(\mathbf{x}) = \begin{cases} \mathrm{Var}\,(\mathbf{x}) & \text{if all components of } \mathbf{x} \text{ are non-negative,} \\ +\infty & \text{otherwise.} \end{cases}$$

With this penalty for possessing a negative component (1,000 in the C code), such portfolios are precluded from being visited by the Markov chain. This is implemented in `NoShorts.cpp`. Again we may take $T = 0$ (if $\mathbf{x}$ and $\mathbf{y}$ are portfolios with no short positions so is $\mathbf{p} = \lambda\mathbf{x} + (1-\lambda)\mathbf{y}$ for $0 \le \lambda \le 1$). The resulting portfolio, with a variance of 4.02856, is given below.

Portfolio composition with no short positions

RTN	7.86
OXY	8.03
F	3.59
UNH	4.15
PG	1.96
UNP	4.36
NKE	10.27
BIIB	1.37
NEE	14.09
WFC	1.85
KHC	2.44
MSFT	2.50
FDX	4.53
VZ	12.05
TGT	3.31
KMI	1.50
LLY	15.57
GD	0.58

As with the no constraints case, more traditional quadratic programming produces virtually the same result.

"Simple" portfolios only. Let us call a portfolio of these 50 stocks "simple" if all the stocks present in the portfolio have equal weight (e.g., 20 stocks with weight $5.00 each). Here we seek the simple portfolio with minimal variance. In this setting quadratic programming methods let us down.

Metropolis Implementation. Here the implementation is different from the previous two applications. We take

$$S = \{\mathbf{x} = (x_1, \ldots, x_{50})^T : \text{each } x_i \in \{0,1\}\},$$

where $x_i = 1$ indicates that the i^{th} stock is present in the portfolio while $x_i = 0$ indicates that it is not. Here $|S| = 2^{50} \approx 1.13 \times 10^{15}$. Of course one "portfolio" in S is not legitimate, namely when all $x_i = 0$. We reflect this in the energy function rather than by deleting it from S. Regarding the notion of neighbor, we'll say that $\mathbf{x} \sim \mathbf{y}$ if they differ at exactly one component. This ensures that the good neighbor rules hold — in particular, each state has $N = 50$ neighbors.

Let $n(\mathbf{x})$ denote the number of stocks present in $\mathbf{x}$. To get the components to sum to 100 (when $n(\mathbf{x}) \neq 0$), we must scale them by a factor of $100/n(\mathbf{x})$, yielding

$$\operatorname{Var}(\mathbf{x}) = \left(\frac{100}{n(\mathbf{x})}\right)^2 \mathbf{x}^T \mathbf{V} \mathbf{x}. \tag{3}$$

We take the energy function to be

$$E(\mathbf{x}) = \begin{cases} \operatorname{Var}(\mathbf{x}) \text{ as in (3)} & \text{if } n(\mathbf{x}) \neq 0, \\ +\infty & \text{if } n(\mathbf{x}) = 0, \end{cases}$$

thus penalizing the "portfolio" $(0, \ldots, 0)^T$ so that the Markov chain will not visit that state.

Here the temperature parameter cannot be taken to be zero or the Markov chain will settle on a stable state that is not optimal. The above argument for zero temperature fails because $\mathbf{p} = \lambda\mathbf{x} + (1 - \lambda)\mathbf{y}$ is not generally a simple portfolio even if $\mathbf{x}$ and $\mathbf{y}$ are simple. A temperature of $T = 0.1$ seems to work well in this setting and the resulting portfolio, with a variance of 4.45182 is given below.

.

Simple portfolio composition

RTN	12.50
OXY	12.50
UNH	12.50
NKE	12.50
NEE	12.50
FDX	12.50
VZ	12.50
LLY	12.50

This is implemented in `Simple.cpp`. Try running the code at zero temperature. It will typically land on a stable state that is not optimal.

§19. Data Clustering

A common task in data analysis involves grouping numerous items of data into smaller clusters having similar features. In the hypothetical example we consider here, 500 high school seniors have taken SAT tests. Their guidance counselor seeks to group them into 20 clusters having similar test scores — presumably for purposes of offering similar advice to students with similar scores. The SAT comprises two tests: Mathematics and Evidence-Based Reading and Writing (Math and EBRW). They are each scored from 200 to 800 in increments of 10. Here an item of data is an ordered pair (x_i, y_i), where x_i is student i's Math score and y_i is the EBRW score. The scatter plot below shows this data for our hypothetical class. The Math average score is 522 and the EBRW average is 530 (typical of the country as a whole) with a correlation of roughly 0.7 (also typical).

K-clustering is a common approach to problems like this. (The K refers to the number of clusters, so here $K = 20$.) Let $\mathcal{D} = \{(x_i, y_i) : 1 \le i \le N\}$ denote the N items of data; here $N = 500$. Partition $\mathcal{D}$ into K non-empty disjoint subsets $\mathcal{D}_k$, $1 \le k \le K$. We seek to minimize the quantity

$$E \;=\; \sum_{k=1}^{K} E_k, \tag{4}$$