To Appear *in the Proceedings of the Second International Conference on Artificial Intelligence Applications on Wall Street, Software Engineering Press, April, 1993, pp. 149-151.*

A Comparison of Stochastic Search Heuristics for Portfolio Optimization

Roy S. Freedman & Rinaldo DiGiorgio
Inductive Solutions, Inc.
New York, NY 10280

Abstract

Modern portfolio theory is based on a rational investor choosing the proportions of assets in a portfolio so as to minimize risk and maximize the expected return. In this paper, we investigate the applicability of different stochastic search heuristics to the problem of finding the optimum portfolio. We compare their performance on two problems with known solutions.

1. Portfolio Optimization

Given a set of assets, what is the optimum proportion of each that is required to achieve an investment objective? Modern portfolio theory is based on a rational investor choosing these proportions so as to minimize risk and maximize the expected return. If risk is measured in terms of the variance of the resulting portfolio, then portfolio optimization is reduced to a "means-variance paradigm" — the optimal portfolio can be derived by knowing the expectations of returns and correlations of returuns for all assets.

Using vector notation from linear algebra, in its simplest form, the portfolio problem is to

Find a vector $\mathbf{w} = (w_1, ...w_n)$
that maximizes the return/risk ratio

$$\frac{[\mathbf{r} \bullet \mathbf{w}]}{(\mathbf{w}^T \bullet \mathbf{C} \bullet \mathbf{w})^{1/2}}$$

subject to constraints

$$m_k = w_k = M_k, \; k=1,...,n$$
$$\text{and } (|w_1| + |w_2| + ... + |w_n|) = 1$$

The vector **w** defines the portfolio: the set of weights that represent the proportion of each asset. Vector **r** is the vector of expected returns for the assets. **C** is a symmetric n-by-n matrix that represents the covariance beteween the returns: c(i,j) is the return covariance between the returns of asset i and asset j:

$$c(i,j) = \sigma_i \, \sigma_j \, \rho_{ij}$$

where ρ_{ij} is the correlation coefficient between the returns for asset i and asset j, and σ_i is the standard deviation of return on asset i. These quantities must be measured statistically. If short sales are allowed in this portfolio then the mimimal constraints mk can be negative. In this case, the absolute value constraints follow the definition of Lintner short sales [1]. Any portfolio **w** that satisfies the constraints is a *feasible solution* to the problem. The goal is to find the best feasible solution.

It is important to note that the above problem can be generalized to include "nonstandard" measures of return or risk, such as geometric return or partial moments, and other types of constraints, such as asset sector limitations.

The portfolio problem represented in the above form is a quadratic programming problem, which can be solved by traditional methods which are variations of the revised simplex linear programming algorithm or gradient-search (hill-climbing) techniques. For some portfolios, these methods may take a long time to solve and, because of the constraints imposed by the problem, may be awkward to initialize to a feasible solution. For example, the simplex algorithm for linear programming exhibits exponential complexity for the worst-case, and polynomial complexity for the average case [2].

Stochastic search algorithms are useful for applications where stable and acceptable (ie, near optimal) answers are desired quickly. In general, stochastic search algorithms do not require knowledge of a derivative (as in gradient-search methods) and perform best with highly nonlinear or highly combinatorial problems. Even though their worst-case behavior is also exponential, their average performance to yield acceptable answers may be quadratic [2]. In this paper, we investigate the applicability of three different heuristics, some having been inspired by biological processes, and compare their performance on two problems with known solutions.

2. Example Problems

Problem A is a simple textbook problem with three assets [1]. There are three assets: the returns and covariance matrix are

		Covariance			
Returns			A1	A2	A3
	14	A1	36	9	18
	8	A2	9	9	18
	20	A3	18	18	225

The constraints are $0 = w_k = 1$ together with the summation equality constraint $\sum w_k = 1$. The theoretical optimum solution is **w** = (14/18, 1/18, 3/18) with Return/Risk ~ 1.66.

Problem B was discussed in [3] and reflects a real example for allocating assets among US and foreign bonds. The returns and covariance matrix for these assets are

		Covariance						
		U.S.	Canada	German	Japan	U.K.	Dutch	French
9.75	U.S.	126	115	64.93	56.7	69	63.5	48.8
10.03	Canada	115	195	94.34	71.7	106	85.4	72.7
9.81	German	64.9	94.3	261.8	179	161	236	201
15.42	Japan	56.7	71.7	178.7	300	154	169	166
12.57	U.K.	69	106	160.9	154	335	149	127
10.48	Dutch	63.5	85.4	236.1	169	149	230	191
10.09	French	48.8	72.7	201.2	166	127	191	200

The constraints are $0 = w_k = 1$ together with the summation equality constraint $\sum w_k = 1$. An optimal solution, computed by traditional methods is **w** = (0.55, 0.0, 0.0, 0.34, 0.09, 0.0, 0.02), with Return/Risk ~ 1.1054.

3. Genetic Algorithm Solution

Genetic algorithms are biologically inspired maximization stochastic heuristics, based on a method that randomly selects two potential solutions from a population of potential solutions, and "breeds" them to create children solutions. A "steady-state" genetic algorithm keeps the population a fixed size: the "worst" solutions in the population are then replaced by the children solutions.

Genetic algorithm techniques and "breeding" operations were originally defined to maximize "fitness" functions of binary arguments. Many of these breeding operations can be extended to other representations, such as integers, real numbers, and permutations.

For the portfolio problem, the population consists of a set of vectors {**w**}, each **w** corresponding to a feasible portfolio. The genetic algorithm operations for real numbers (based on [4]) are crossover, coarse tuning, fine tuning, average, and mutation, each with fixed operator probabilities and fixed probabilities that are used to determine which set of weights that the operator will be applied. The operators are adjusted so that they generate a random feasible solution. After each iteration, one or more of the breeding operators are selected, and applied to a two randomly selected parents, whose selection probability is proportional to the Return/Risk ratio that is to be maximized (the "fitness" or objective function).

Problems A and B were both solved using GenSheet, a commercially available genetic algorithm package. For Problem A, the population started converging to the optimal solution (to two decimal place accuracy) in 10 generations. For Problem B, after generating the following randomly generated population...

U.S.	Canada	German	Japan	U.K.	Dutch	French	**RET**	**RISK**	Ret/Risk
0.06	0.09	0.185	0.2	0.21	0.12	0.14	11.628	12.748	0.9122
0.11	0.03	0.193	0.09	0.17	0.15	0.24	10.992	12.626	0.8705
0.06	0.21	0.084	0.26	0	0.18	0.2	11.483	12.152	0.9449
0.03	0.04	0.058	0.22	0.34	0.25	0.05	12.197	13.489	0.9042
0.16	0.11	0.129	0.16	0.15	0.11	0.17	11.263	11.755	0.9581
0.18	0.01	0.226	0.19	0.17	0.12	0.11	11.438	12.392	0.923
0.1	0.26	0.067	0.23	0.26	0.07	0.02	11.913	12.02	0.9911
0.19	0.16	0.041	0.18	0.26	0.11	0.06	11.665	11.65	1.0013
0.19	0.1	0.092	0.01	0.27	0.09	0.24	10.76	11.709	0.919
0.1	0.15	0.235	0.18	0.01	0.06	0.28	10.96	12.262	0.8939
0.19	0.05	0.2	0.22	0.2	0.14	0.01	11.687	12.206	0.9575
0.07	0.24	0.209	0.2	0.2	0	0.08	11.527	12.058	0.9559

...the population started converging to the optimal solution (to two decimal place accuracy) in 100 generations:

U.S.	Canada	German	Japan	U.K.	Dutch	French	**RET**	**RISK**	Ret/Risk
0.56	0	0	0.27	0.17	0	0	11.776	10.751	1.0954
0.53	0.03	0	0.32	0.03	0	0.09	11.692	10.638	1.099
0.56	0.01	0	0.25	0.14	0.03	0.02	11.562	10.57	1.0938
0.56	0.01	0	0.24	0.16	0	0.03	11.582	10.597	1.0929
0.6	0.02	0	0.26	0	0.09	0.03	11.311	10.459	1.0814
0.48	0.04	0	0.23	0.14	0	0.11	11.522	10.551	1.092
0.6	0.02	0	0.26	0.1	0	0.03	11.506	10.486	1.0973
0.6	0.02	0	0.26	0.1	0	0.03	11.506	10.486	1.0973
0.6	0.02	0	0.26	0.1	0	0.03	11.506	10.486	1.0973
0.48	0.04	0	0.23	0.14	0	0.11	11.522	10.551	1.092
0.48	0.04	0	0.23	0.14	0	0.11	11.522	10.551	1.092
0.56	0.01	0	0.25	0.14	0.03	0.02	11.562	10.57	1.0938

4. Simulated Annealing

The Metropolis algorithm [2] simulates the evolution of a physical system in contact with a heat-bath as it is observed at random times. Lowering the temperature results in freezing the randomized behavior of the physical system, so it approaches a ground or "annealed" state. In operation, the Metropolis algorithm assumes an "energy" function f(x) is defined on a state x. We generates a new state x_i from an old state x_j randomly, and accept the new state if $f(x_j) - f(x_i) = 0$; otherwise we accept x_i with probability P, where

$$P = \exp(- [f(x_j) - f(x_i)]/T)$$

This expression is based on a heuristic derived from the Boltzmann probability distribution of thermodynamics. As T approaches zero, the "randomization" becomes frozen: the optimal final state is reached. In the Metropolis simulated annealing algorithm, this is accomplished with an "annealing schedule" that for each iteration, generates a sequence T_k that converges to zero. At zero temperature, there are no random solutions generated.

For the portfolio problem, the simulated annealing states correspond to feasible portfolios. The function f is just the Return/Risk ratio. At each iteration, the algorithm randomly selects a set of asset weights to change, and generates a random feasible solution. The initial fixed temperatures updated by the annealing schedule T := .9*T.

After each iteration, new solutions are evaluated and accepted with respect to the Boltzmann probability distribution. For Problem A, the portfolio converging to the optimal solution (to two decimal place accuracy) in 56 iterations. For Problem B, the portfolio started converging to the optimal solution (to two decimal place accuracy) in 379 iterations.

5. Dynamic Search Space Reduction

This heuristic finds the optimal solution to a nonlinear optimization problem by reducing the size of the search region dynamically. After generating a random feasible solution **w**, the algorithm, generates a new vector $\mathbf{w}_{new}$ from a random number R, with

$\mathbf{w}_{new} = \mathbf{w} + \mathbf{g}$

where $\mathbf{g} = [\mathbf{M} - \mathbf{m}]*R$

and $\mathbf{M} = (M_1, ..., M_n)$,

and $\mathbf{m} = (m_1, ..., m_n)$.

If $\mathbf{w}_{new}$ is not feasible then a new $\mathbf{w}_{new}$ is generated.

For Problem A, the portfolio converging to the optimal solution (to two decimal place accuracy) in 18 iterations. For Problem B, the portfolio started converging to the optimal solution (to two decimal place accuracy) in 115 iterations.

6. Discussion

Simulated annealing was the poorest performer of the three stochastic search methods for the portfolio problem. This may be because of too much "randomization" required in the algorithm: the weights are too sensitive to randomized changes because of the constraints. The genetic algorithm and dynamic search space reduction algorithm had similar performance. Both heuristics are similar and are more sensitive to the constraints. Because of its emphasis on population convergence, the genetic algorithm identified alternate solutions better than the other two algorithms.

In general, the difficulty in all optimization techniques concerns the constraints, ie, in generating a feasible solution. Stochastic search techniques seem to live up to their promise in generating acceptable solutions quickly, even with complicated constraints and objective functions.

For Problem B, both algorithms took less than a second to execute. Informal studies indicate that for larger portfolios (200 to 1000 assets), execution times for acceptable non-optimal solutions scale up quadratically. We are now investigating parallel implementations of stochastic search algorithms for very large portfolios.

7. References

1. Elton, E., Gruber, M., 1991. *Modern Portfolio Theory and Investment Analysis*. John Wiley: New York p. 69.

2. Bonomi, E., Lutton, J., 1984. The N-City Traveling Salesman Problem: Statistical Analysis and the Metropolis Algorithm. *SIAM Review, 26:4, pp. 551-568*.

3. Jorion, P., 1992. Portfolio Optimization in Practice. *Financial Analysts Journal*, January-February, pp. 68-74.

4. Davis, L., 1989. Adapting Operator Probabilities in Genetic Algorithm. In *Proceedings of the Third International Conference on Genetic Algorithms*, Morgan Kaufmann: San Mateo, pp. 61-69.

5. Guggenheimer, H. 1989. *Lecture Notes in Numerical Analysis*, Polytechnic University.