X Congresso Brasileiro de Inteligência Computacional (CBIC'2011), 8 a 11 de Novembro de 2011, Fortaleza, Ceará

TOPOLOGICAL AND MOTION STRATEGIES FOR CELLULAR MEMETIC TREE-BASED PORTFOLIO OPTIMIZATION

Claus Aranha
Electrical Engineering Program, COPPE, Federal University of Rio de Janeiro
claus.aranha@lps.ufrj.br

Carlos R. B. Azevedo
Laboratory of Bioinformatics and Bioinspired Computing (LBIC), FEEC, University of Campinas
azevedo@dca.fee.unicamp.br

V. Scott Gordon
California State University Sacramento
gordonvs@ecs.csus.edu

Hitoshi Iba
Institute of Electrical Engineering, The University of Tokyo
iba@iba.t.u-tokyo.ac.jp

Resumo – A Otimização de Portfólios (OP) é um problema de alocação de recursos, no qual pesos são associados a vários produtos financeiros de maneira a maximizar o lucro e minimizar o risco. O Algoritmo Memético Baseado em Árvores (MTGA), usando uma combinação de representação por árvore binária e busca local (LS), mostrou ótimo desempenho neste problema quando comparado com outras técnicas de otimização de pesos. Neste artigo nós estudamos quatro implementações híbridas entre o MTGA e técnicas topológicas, denominadas como Algoritmos Meméticos Celulares (CMA). A diferenças entre as versões híbridas são as políticas de movimento dos indivíduos e distribuição dos meta parâmetros de auto adaptação. As quatro abordagens são comparadas usando simulações por dados históricos. A melhor performance é atingida por um CMA com política de movimento, enquanto que a auto configuração de parâmetros não obteve tanto sucesso. Os resultados obtidos não apenas sugerem um melhor método para o problema de OP, mas também mostram novos ramos de investigação para modelos celulares.

Palavras-chave – Algoritmos Meméticos, Auto-adaptação, Modelos Baseados em Terreno, Hibridização, Engenharia Financeiras, Otimização de Portfólios

Abstract – Portfolio Optimization (PO) is a resource allocation problem where real valued weights are assigned to multiple financial assets in order to maximize the return and minimize the risk. The Memetic Tree-based Algorithm (MTGA), employing a tree representation allied with local search (LS) has shown great performance compared to other weight balancing techniques. In this work, we hybridize MTGA with topological frameworks — Cellular Memetic Algorithms (CMA) — and study four implementations, varying whether the individuals move through the grid, and whether meta-parameters are spread along the axes for self-adaptation. The approaches are compared using a historical data simulation. A CMA which incorporates motion performs best, while parameter tuning was less successful. The results not only describe an improved method for PO, but also have broader implications for cellular models wherein the benefits of motion are shown to deserve further investigation.

Keywords – Memetic algorithms, self-adaptation, terrain-based models, hybridization, financial applications, portfolio optimization

1. INTRODUCTION

Portfolio Optimization (PO), an important problem in Financial Engineering, consists of dividing an amount of capital between assets to maximize the return and minimize the risk of the investment. Mathematically, PO can be modeled as a Resource Allocation Problem, where the resource is the investment capital, the jobs to which the resource is assigned are the assets, and the utility functions are the risk and the expected return of the investment.

Investment Portfolios are used in long term management of savings accounts, retirement funds, among others. The idea is that by investing in multiple counter-correlated assets at the same time, it is possible to reduce the overall risk of the investment.

While the mathematical model of PO can be solved by programmatic optimization methods, real world instances with large data sets and many constraints are too complicated to be solved in this way. Many works have shown that Evolutionary computation is a good alternative in these cases [1–3]. In particular, the Memetic Tree-based Algorithm (MTGA), composed of a binary tree representation and local search, has been successful in calculating well-performing portfolios for markets with a large number of assets [4].

In this work, we aim to improve the MTGA's performance by adding a topology framework to the algorithm. First, we incorporate a cellular strategy, the Cellular Memetic Algorithm [5] (CMA). The CMA, a diffusion technique, places individuals on a two-dimensional grid and limits their breeding to between neighbors. Such strategy encourages the formation of niches, enhancing diversity [6].

Next, the Terrain Based Memetic Algorithm (TBMA) [7] is used to explore the self-adaptation of two MTGA parameters. The TBMA has been first defined as a CMA in which the behavior of the local search (LS) depends on the topological distribution of its parameter values over the cellular grid. This allows one population to benefit from a variety of available combinations of LS parameter settings.

The "motioner" TBMA (mTBMA) extends the idea by enabling individuals to move around the grid. Individuals will prefer to stay in grid cells with the best parameter values. The evaluation of the parameter values is based on the fitness of individuals already in that grid cell. The mTBMA has previously shown very good performance on the image vector quantization problem [7].

We break this framework into two components: the *motion strategy* and the *terrain strategy* (see figure 1). The motion strategy dictates whether or not individuals can move around the grid. The terrain strategy deals with whether MTGA parameters are distributed on the grid to achieve self-adaptation. The combination of the two components leads to four strategies: static Cellular, static Terrain-Based, motioner Cellular, and motioner Terrain-Based.

To our knowledge, the "motioner Cellular" strategy has not yet been assessed in existing works. To date, the motivation for adding motion to a cellular model has been to address known limitations of static Terrain-Based Strategies [8], but not for the sake of motion itself.

We investigate those four variations by analyzing the effects of both the terrain strategy and the motion strategy separately and combined. We then find which combination improves MTGA performance the most. We compare these variants with the pure MTGA and two benchmarks in multiple data sets for the Portfolio Optimization Problem.

2 Problem Description

The Portfolio Optimization problem consists of choosing the optimal combination of financial assets in a given investment, in order to maximize its return while minimizing its risk. The main idea is that if you invest in two counter correlated assets, their risks cancel each other out, resulting in a Portfolio with a smaller amount of total risk for the same return.

A mathematical model for the PO problem was proposed by Markowitz [9], in which a portfolio P is defined as a set of N real numbers $(w_0, w_1, ...w_N)$ corresponding to the weights of the N available assets in the market. These weights must obey two basic restrictions: The sum of all weights w_0 to w_N must be equal to 1, and the value of each weight w_i must be between 0 and 1.

Each asset has an expected return value, expressed by R_i. The expected return value for the portfolio R_P is given by the sum of the expected return values for the assets that are part of that portfolio, $(R_P = \sum R_i w_i)$.

Also, each asset has a risk measure, σ_i. In the Markowitz model, the risk of an asset is defined as the variance of that asset's returns over time, and the risk of the Portfolio is defined as the covariance between its assets. The risk of the Portfolio is calculated as

$$\sigma_P = \sum_{i=0}^{N} \sum_{j=0}^{N} \sigma_{ij} w_i w_j, \tag{1}$$

where $\sigma_{ij}, i \neq j$ is the covariance between i and j. If $i = j$, σ_{ii} is the variance of asset i. These two utility measures can be combined to form the *Sharpe Ratio*, which is often used in the financial field to evaluate investments. The Sharpe Ratio is calculated as

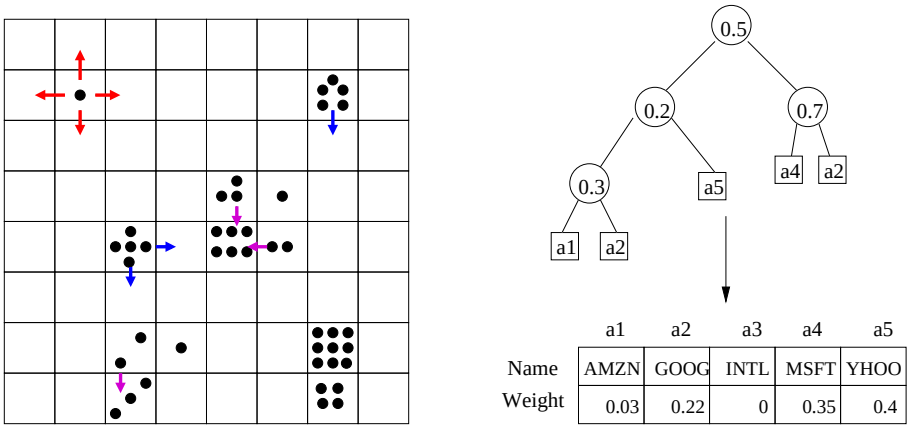
$$Sr = \frac{R_P - R_{riskless}}{\sigma_P}, \tag{2}$$

where $R_{riskless}$ is the risk-free rate, an asset with 0 risk and a fixed, low return rate (for example, government bonds of stable nations). The Sharpe Ratio expresses the trade-off between risk and return for a Portfolio given a fixed rate of return. A higher Sr value indicates a better Portfolio.

2.1 Evolutionary Approaches

A very common way to address the PO Problem is by means of Weight Optimization. In this strategy, the system will try to simultaneously determine the best weight for each of the assets. This is most commonly achieved by a Genetic Algorithm (GA) where the genome is an array (A) of the assets weight in the portfolio (e.g. $[2, 10]$). In this GA, A stores as many elements as there are assets in the market. Each element a_i is a real value which defines the exact weight of that particular asset in the portfolio.

However, it is difficult for a standard GA to fine-tune real valued genomes without specialized crossover operators [11]. Furthermore, the array representation does not include information about the covariance between the different assets, which makes the search blind to an important source of information in the utility function. These issues have been addressed by the usage of binary arrays for asset selection [12] and by the random selection of a subset of the market assets to form the array [3]. The Memetic Tree based GA (MTGA) instead eschews the array altogether by representing the portfolio as a binary tree, where the leaf nodes are the assets and the intermediate nodes are relative weights between its sub-trees. This allows for information



	a1	a2	a3	a4	a5
Name	AMZN	GOOG	INTL	MSFT	YHOO
Weight	0.03	0.22	0	0.35	0.4

Figure 1: The TBMA grid, and a Tree genome and its portfolio

about the relationship of the assets to be stored, and LS functions to be used in an efficient manner. The MTGA has been shown to outperform array based representations [4].

Besides weight optimization, another way to address the PO problem is by equally-weighted asset selection. In this strategy, the market assets are analyzed and the best ones are selected to take part in the portfolio. For instance Hassan and Clack use Genetic Programming rules, including indicators of financial performance, to decide whether to include or exclude an asset in a portfolio with a limited number of open "slots" [1].

3 The MTGA

The MTGA uses a binary tree representation, which allows for a "divide-and-conquer" approach to local search, where a hill climbing algorithm is applied from the bottom nodes towards the root, to calculate the best local weight for each node without taking into account all assets at the same time.

In this representation, each non-terminal node holds a relative real valued weight w_i ($0 \leq w_i \leq 1$) between its two sub-trees. The left sub-tree's weight is defined as w_i, and the right sub-tree's weight as $1 - w_i$. Each terminal node holds the index of an asset in the market. It is possible to have more than one terminal pointing to the same asset in the same tree. Figure 1 shows this representation.

To obtain the portfolio, we calculate the weight of each terminal node by multiplying the weights of all nodes that need to be visited to reach that terminal, starting from the root of the tree. After all terminal nodes are visited, the weights of those terminals that point to the same asset are added together. The assets which are not pointed to by any terminal node in the tree are given weight 0. This process is detailed in Algorithm 1

Algorithm 1 getPort - Extracts portfolio from genome

Require: initial tree node n.
  if n is a terminal node **then**
    i is the index value of n
    Set weight of index i $w_i = 1$
    Create Portfolio P and add w_i to it
    return Portfolio P
  else
    n_l is the left child of n; n_r is the right child of n
    Portfolio L = getPort(n_l); Portfolio R = getPort(n_r)
    W is the weight value of n
    Create Portfolio P
    for Each asset i **do**
      $w_i^p = (W)w_i^l + (1 - W)w_i^r$
    end for
    return Portfolio P
  end if

3.1 Crossover, Mutation and Local Search Operators

The mutation and crossover in MTGA are standard tree operators, the first generating a new tree from a randomly selected node, and the second exchanging sub-trees between the selected parents. The cut-off node for these operators is selected by choosing a random depth in the target tree, and following a random path from the root to this depth.

Algorithm 2 Local Search

Require: Child nodes are leaves or locally optimized.
Ensure: Current node is locally optimized
  while $(|meme\_speed| > meme\_tresh)$ AND $(0 < weight < 1)$ **do**
    $old\_fitness = fitness$
    $weight = weight + meme\_speed$
    if $weight > 1$ **then**
      $weight = 1$
    end if
    if $weight < 0$ **then**
      $weight = 0$
    end if
    calculate_fitness($weight$)
    if $fitness < old\_fitness$ **then**
      $meme\_speed = meme\_speed * meme\_accel * -1$
    end if
  end while

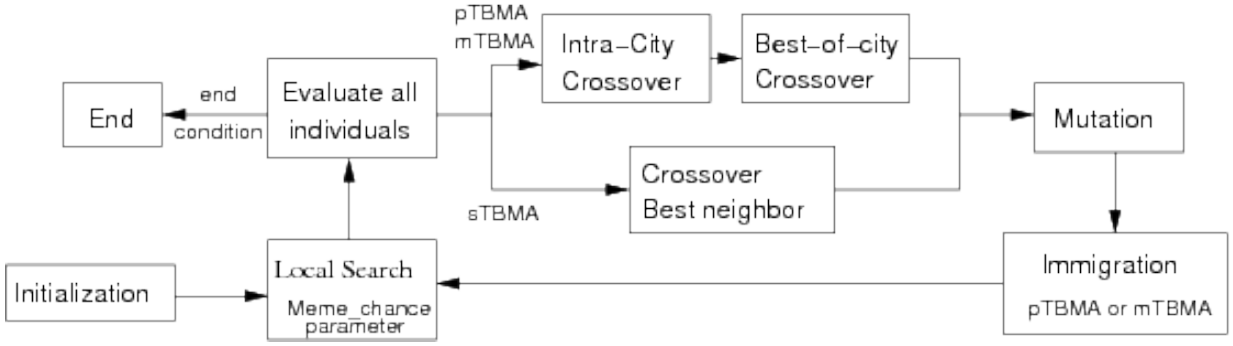


Figure 2: The flowchart of the MTGA-TBMA hybrid

The local search operator aims to finely tune the intermediate weights in the tree. It complements the structural search done by the crossover and mutation operators. At the start of every generation, a number of individuals are chosen, and the local optimization step is applied to them. Fitness evaluation and selection is only realized after the local optimization step.

For each selected individual, the local search operator starts a recursive hill climbing optimization. Starting from the root node, it recursively leads down to the deepest level of the tree (first the left sub-tree, then the right). There, hill-climbing is applied to the weight, and the risk and return values of the two-asset portfolio in that node are calculated, following algorithm 2. After all intermediate nodes in the bottommost level have been calculated in this way, these values are used to calculate the two asset portfolio for the nodes one level above, and so on until the program returns to the root node.

Hill climbing is shown in algorithm 2, where the parameter *meme_speed* represents the amount by which the weight is incremented (or decremented) during hill-climbing. *meme_accel* is the value by which *meme_speed* decreases every time the weight crosses the optimal point. These two values thus determine the size of the steps taken at each iteration. The value of *meme_accel* must be $0 <$ *meme_accel* < 1, and the value of *meme_speed* should be a small fraction of the range of the variable being optimized. The parameter *meme_tresh* is the minimum value of *meme_speed* which signals the end of hill climbing. The search also ends if the weight reaches 0.0 or 1.0 (when the optimal point is not in the weight range $[0, 1]$).

4 Topological Implementation

In this research, we aim to improve the performance of the MTGA by incorporating a topological framework. We implement and test four different versions, which we classify according to their *Motion Strategy* and their *Terrain Strategy*. The interface between the MTGA and the TBGA is illustrated in Figure 2.

The Motion Strategy determines the relationship between individuals and their position in the grid. In the *Static* implementation, each cell in the grid holds a single individual. Mating and Selection is based on a fixed neighborhood around an individual. In the *Motioner* implementation, individuals can migrate between cells, and a cell may contain multiple individuals. Selection and Crossover typically happen among individuals in the same cell.

Terrain refers to the assigning of different combinations of parameter values (mutation rate, for example) to each grid cell, allowing the algorithm to exploit different parameter settings at different times.

Our classification gives rise to four versions, most of which have been studied previously: (1) The Stationary+Cellular strategy corresponds to the Cellular Memetic Algorithm (CMA) [5]. (2) The Stationary+Terrain strategy corresponds to the Terrain-Based

Memetic algorithm (TBMA) described by Azevedo et al. [13]. (3) The Motion+Cellular strategy is dubbed mCMA, which we believe to be previously untested and thus novel to this work, and (4) The Motion + Terrain strategy, first described by Krink and Ursem as the Patchwork Model [8], and later adapted for image vector quantization by Azevedo and Gordon [7]. In versions (3) and (4), motion was added to the terrain-based model in order to enable more individuals to exploit regions of the grid with good parameters.

For clarity, we will call each version: (1) CMA, (2) TBMA, (3) mCMA (motioner-CMA) and (4) mTBMA (motioner-TBMA).

4.1 Stationary Strategy Implementation

The Stationary Strategy follows the basic concept of a diffusion GA [14], with individuals occupying a toroidal 2D grid. At the initialization stage, one individual (tree) is created in each cell. After that, for every generation, each individual T_i selects the most fit from the four neighbors around it and generates a new individual, T_i', through crossover. After all individuals have generated an offspring, the offspring are evaluated, and those who outperform their main parent's fitness replace them in the cell.

4.2 Motioner Strategy Implementation

Each individual in the initial population is assigned to a random cell. Thus, cells are likely to contain different numbers of individuals. A cell containing at least one individual is called a *city*.

At every generation, three steps are made. First, for each city with only one individual, mutation and local search operators are applied, and the individual moves to a random neighbor cell (red arrows in Figure 1). For cities with more than one individual, a number of crossover operations equal to the population of the city are performed, each followed by mutation and local optimization. For each crossover operation, the offspring generated replaces the parent with the lowest fitness.

In the second step, the fittest member of each city i – called the *mayor* (M_i) – selects the neighboring mayor with highest fitness (if any exists), and undergoes crossover. The offspring replaces the mayor if it has a higher fitness value.

In the third step, every individual migrates to a neighboring city if that city contains a higher fitness mayor (purple arrows in Figure 1). If there is no such neighboring city, migration may occur anyways, with probability $p_{migration}$ (blue arrows in Figure 1).

The rules for migration and crossover aim to balance the goals of clustering individuals around parameters which are yielding the best solutions, while guaranteeing that the population will try a variety of parameter values. For further details, see [7, 13]

4.3 Terrain Strategy Implementation

In the Terrain Strategy, each individual utilizes parameter settings corresponding to its cell. In order to preserve a gradual change of parameters at all points in the toroidal grid, we used an assignment method called "sifting" [14], in which values increase in an alternate left-right pattern along a row (or column). For instance, if the range of parameter valuesthe parameter domain is $[0, 10]$, the actual distribution along a row would be $(9, 7, 5, 3, 1, 0, 2, 4, 6, 8, 10)$.

In this work, we selected *Tree Depth* and *Pruning Policy* as the terrain variables. Tree depth controls the maximum depth of a PO solution. Pruning Policy determines whether to remove sub trees when an intermediate weight is set to the values of 0 or 1 by the local search, which would simplify trees but possibly remove inactive building blocks from the population.

5 Experiments

We use a market simulation based on historical prices to compare the performance of the four topology frameworks. We compare their results with the pure MTGA and two benchmarks (DEahcSPX [15] and Market Index). The returns and risk of each asset are calculated based on the monthly closing prices of the assets. We set the return of the riskless asset at 3%, and the trading costs at 2%, values chosen based on discussions with traders.

5.1 Experiment Setup

We use two data sets in our simulations: The NASDAQ data set, and the S&P500 data set. The NASDAQ data set contains 100 assets, and the S&P has 500. The expected return for each month is calculated as the moving average of the returns from the previous 12 months. The data sets contain historical data up to December 2008, which were obtained from freely available on-line sources[1].

There are 36 scenarios, each composed of the past return, expected return, and correlation information of a single month, from January 2006 to December 2008 (36 months). With two data sets, we have a total of 72 different cases.

The evolutionary systems used in the experiments had the following values for their global parameters. For the general evolutionary parameters, we have used the following parameters: 300 generations, 300 individuals, crossover rate 0.8, and mutation rate 0.03. These parameters were chosen by tuning on a separate testing data set. The results presented here are on the validation data set.

For the parameters specific to the MTGA, the tree depth was set to $\lceil log_2 D \rceil$ where D is the dimensionality of the problem. For the local search operator, we use 0.1 for meme_speed, 0.333 for meme_accel, and 0.003 for meme_tresh. We found that the parameters for this local search do not heavily change the results, as long as they are within the previously defined boundaries.

[1] http://finance.yahoo.com

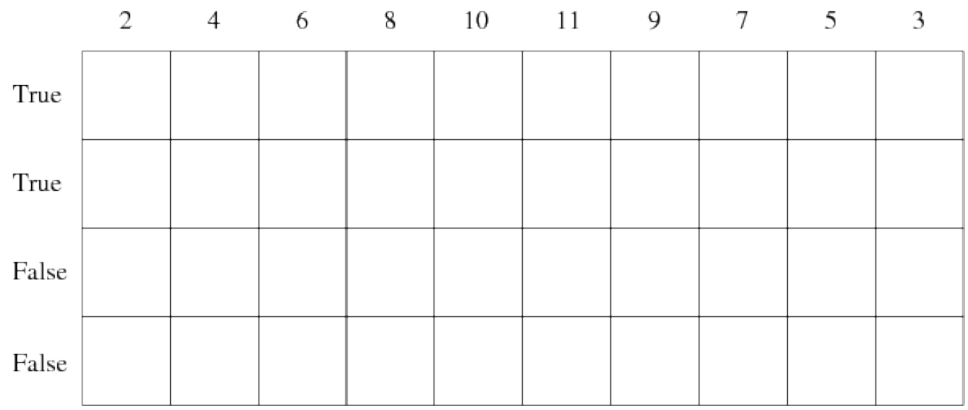


Figure 3: The 2-D grid used by the CMAs in our experiments

Table 1: Sharpe Value for some of the 72 scenarios in the NASDAQ and S&P data sets. These are averages of the best individual of 20 runs. For numbers in Italics, their difference is not statistically significant with relation to the MTGA at a 0.05 level.

Date NASDAQ	MTGA	CMA	mCMA	mTBMA	TBMA	DEahcSPX	Index
Feb. 2006	2.1142	*2.1051*	2.1293	1.2956	*2.0917*	1.5667	-0.49
May. 2006	2.1485	*2.2208*	2.4281	1.9045	*2.1494*	1.5733	-0.21
Aug. 2006	1.3522	*1.3838*	*1.3723*	1.1809	*1.3691*	0.9662	-0.68
Nov. 2006	0.9035	0.9139	0.9166	0.8490	*0.9116*	0.6963	-0.53
Feb. 2007	1.3606	*1.3611*	*1.3620*	1.2757	*1.3625*	1.0521	-0.68
May. 2007	0.9339	*0.9347*	0.9350	0.9284	*0.9342*	0.7969	-0.33
Aug. 2007	6.1738	*7.2531*	*6.2943*	3.4901	*6.6812*	2.5784	-0.90
Nov. 2007	4.638	*5.0162*	*5.0089*	2.1994	*5.1285*	1.8283	-0.30
Feb. 2008	0.5356	0.5356	*0.5356*	*0.5262*	*0.5356*	0.4596	-0.49

Date S&P500	MTGA	CMA	mCMA	mTBMA	TBMA	DEahcSPX	Index
Feb. 2006	12.5015	*10.1107*	*13.1852*	3.1820	*10.3171*	2.0617	-1.13
May. 2006	9.7257	*10.8763*	14.3320	3.2343	*10.3777*	1.8598	-0.88
Aug. 2006	1.2239	*1.2498*	*1.2414*	1.0746	1.1971	0.5196	-0.54
Nov. 2006	4.7003	*4.7508*	5.4673	1.7197	*4.2404*	0.9648	-1.05
Feb. 2007	5.4636	*6.0573*	6.6911	1.8469	*5.0948*	0.7403	-1.27
May. 2007	4.3045	*5.5955*	*4.1455*	2.0035	*4.0515*	0.1383	-1.16
Aug. 2007	36.0465	20.6902	*36.1979*	5.1093	*24.7126*	0.6763	-1.08
Nov. 2007	11.5642	*11.0028*	14.8134	3.4904	*10.1393*	1.0450	-0.87
Feb. 2008	0.9033	0.9139	0.9161	0.8683	*0.9062*	0.5680	-1.01

6 Topological Implementation

Our grid is a 10x4 rectangle, as shown in Figure 3. The size of the retangle is defined by the number of parameter values that we are interested in testing. In both the TBMA and mTBMA, we assign values for the Tree Depth parameter from 2 to 11, along the x-axis. For pruning policy, a "true" value is assigned to the two top rows, and "false" to the two bottom rows. These values indicate whether or not the individual should undergo pruning. In both the mCMA and mTBMA, each cell holds a maximum of 150 individuals, and the migration probability ($p_{migration}$) is 0.2.

6.1 Results

In this experiment we generate portfolios for the 72 scenarios, and compare the achieved Sharpe Ratio values. The MTGA is our baseline method. We compare the MTGA with the four algorithms discussed in this paper: The CMA, the mCMA, the TBMA, and the mTBMA.

The results for these comparative experiments are displayed in Table 1, which shows a sample of the results from the 72 scenarios. In total, the four frameworks employing topological methods were superior to the MTGA with a statistically significant difference in 25 of the 36 cases in the NASDAQ data set, and 33 of the 36 cases in the harder S&P data set. In particular, the mCMA was statistically better than all other methods in 6 of 36 cases in the NASDAQ data set, and 12 of 36 cases in the S&P data set, the highest score among the four methods.

7 Discussion

As shown in Table 1, the MTGA-based methods outperform the DEahcSPX and the Market Index overwhelmingly, suggesting that our proposed tree representation for PO and its associated local search method provides a powerful evolutionary framework for quickly removing and adding completely new groups of assets, facilitating fast exploration of the search space.

Results for hybrid methods which incorporate topological structure show further improvement. The best of these, mCMA, outperforms the MTGA in 51 of the 72 cases ($\approx 71\%$), sometimes with statistical significance ($\approx 33\%$). Some of the other hybrid configurations also perform well, frequently besting MTGA.

Our attempts at utilizing terrain for fine-tuning the MTGA parameters was less successful. Although the terrain-based Memetic algorithm (TBMA) did very well on the NASDAQ data set, adding motion to the TBMA did not have the positive results that had been observed in earlier studies. It is possible that our method of pruning individuals whenever they move towards a cell with lower Tree Depth value prevents them from maintaining good building blocks.

While it is not surprising that a cellular algorithm would perform well, it is interesting that the best performing framework is the previously untested motioner cellular model. As described earlier, motion has in the past only been added to terrain-based models to contravene a known deficiency in those configurations. We implemented the mCMA for completeness due to the ontology we chose for our experiments. We are unaware of any other motivation for adding motion to a cellular GA.

Speculation on the reason for mCMA's superior performance is arguably premature. But we can suggest two possibilities: (1) as cities form, locality decreases, so perhaps an initially high but gradually decreasing degree of locality has some advantage for this sort of application, gradually intensifying the search through the generations; or (2) the concept of high-fitness individuals competing to attract others to their neighborhood strengthens the mCMA, as occurs for example in the Hierarchical CGA [16], albeit using a different mechanism than ours.

8 Conclusion

We have described a system for evolutionary Portfolio Optimization which combines a flexible tree-based representation, a powerful associated local search method, within a topological framework utilizing a variety of configurations of cellular Memetic algorithms (CMA). The system, which we called a hybrid MTGA, was then tested using historical data from NASDAQ and S&P indexes.

The hybridization with CMA generally boosted MTGA's performance. However, our attempts at tuning its parameters with a terrain-based approach (TBMA and mTBMA) did not perform as well. The best performing framework was a novel cellular Memetic algorithm with motion (mCMA).

The results raised unexpected and potentially significant questions as to the role of locality in cellular genetic search. While the traditional cellular model outperformed the global-population based prior approach, allowing individuals in the cellular grid structure to move, thereby reducing their locality over time, resulted in dramatically improved results. While it is possible that this may be an isolated incident, our first experience with this simple population model indicates that the mCMA is worthy of broader testing.

References

[1] G. Hassan and C. D. Clack. "Multiobjective Robustness for Portfolio Optimization in Volatile Environments". In *GECCO'08*, pp. 1507–1514, Atlanta, Georgia, 2008.

[2] C.-M. Lin and M. Gen. "An Effective Decision-Based Genetic Algorithm Approach to Multiobjective Portfolio Optimization Problem". *Applied Mathematical Sciences*, vol. 1, no. 5, pp. 201–210, 2007.

[3] P. Lipinski, K. Winczura and J. Wojcik. "Building Risk-Optimal Portfolio Using Evolutionary Strategies". In *EvoWorkshops 2007*, edited by M. G. et al., number 4448 in LNCS, pp. 208–217. Springer-Verlag, 2007.

[4] C. Aranha and H. Iba. "The Memetic Tree-based Genetic Algorithm and its application to Portfolio Optimization". *Memetic Computing*, vol. 1, no. 2, pp. 139–151, June 2009.

[5] N. Huy, O. Soon, L. Hiot and N. Krasnogor. "Adaptive cellular memetic algorithms". *Evolutionary Computation*, vol. 17, no. 2, pp. 231–256, 2009.

[6] V. S. Gordon and L. D. Whitley. "Serial and Parallel Genetic Algorithms as Function Optimizers". In *Proceedings of the 5th International Conference on Genetic Algorithms*, pp. 177–183, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

[7] C. R. Azevedo and V. S. Gordon. "Adaptive terrain-based memetic algorithms". In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation - GECCO '09*, pp. 747–754, New York, New York, USA, 2009. ACM Press.

[8] T. Krink and R. Ursem. "Parameter Control Using the Agent Based Patchwork Model". In *Proceedings of the 2nd Congress on Evolutionary Computation*, pp. 77–83, 2000.

[9] H. Markowitz. *Mean-Variance analysis in Portfolio Choice and Capital Market*. Basil Blackwell, New York, 1987.

[10] R. Hochreiter. "An Evolutionary Computation Approach to Scenario-Based Risk-Return Portfolio Optimization for General Risk Measures". In *EvoWorkshops 2007*, edited by M. G. et al., number 4448 in LNCS, pp. 199–207. Springer-Verlag, 2007.

[11] B. Ullah, R. Sarker, D. Cornforth and C. Lokan. "An Agent-based Memetic Algorithm (AMA) for Solving Constrained Optimization Problems". In *IEEE Congress on Evolutionary Computation (CEC)*, pp. 999–1006, Singapore, September 2007.

[12] P. Skolpadungket, K. Dahal and N. Harnpornchai. "Portfolio optimization using multi-obj ective genetic algorithms". In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pp. 516–523, 2007.

[13] C. R. Azevedo, F. Azevedo, W. Lopes and F. Madeiro. *Terrain-based memetic algorithms for vector quantizer design*, volume 236 of *Studies in Computational Intelligence*, chapter 17, pp. 197–211. Springer, Berlin, Heidelberg, 2009.

[14] V. S. Gordon, R. Pirie, A. Wachter and S. Sharp. "Terrain-Based Genetic Algorithm (TBGA): Modeling Parameter Space as Terrain". In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '99)*, pp. 229–235, Orlando, FL, USA, 1999. Morgan Kaufmann Publishers Inc.

[15] N. Noman and H. Iba. "Accelerating Differential Evolution Using an Adaptive Local Search". *IEEE Trans. Evolutionary Computation*, vol. 12, no. 1, pp. 107–125, 2008.

[16] S. Janson, E. Alba, B. Dorronsoro and M. Middendorf. "Hierarchical cellular genetic algorithm". In *Evolutionary Computation in Combinatorial Optimization EvoCOP06*, volume 3906, p. 111, Berlin, Heidelberg, 2006. Springer.