3-11-2002

Learning Linear Causal Structure Equation Models with Genetic Algorithms

Shane Harwood
Carnegie Mellon University

Richard Scheines
Carnegie Mellon University

Learning Linear Causal Structure Equation Models with Genetic Algorithms

Shane Harwood and Richard Scheines

March 11, 2002

Technical Report No. CMU-PHIL-128

Learning Linear Causal Structure Equation Models with Genetic Algorithms

Shane Ellis Harwood and Richard Scheines
Carnegie Mellon University, Pittsburgh, PA, USA 15232
harwood@andrew.cmu.edu

Abstract

In this paper, we present an efficient genetic algorithm that searches for causal relationships in linear systems. A Directed Acyclic Graph (DAG) coupled with an appropriate parameterization yields a linear Structural Equation Model (SEM). Graphs are explored based on an evolving genetic population, with mutations, crossover, and fit determined by a score based on the Bayesian Information Criterion (BIC), but that varies the standard penalty associated with the quantity of edges. The parameters of search include: scoring method, mating behavior (e.g. mutation rate and quantity of offspring), method of population selection, and a small set of heuristics encoded by Boolean variables. The settings of these parameters describe multiple possible search strategies and can have a large effect on performance. Multiple genetic searches with varying parameterizations are performed on a set of simulated data. The set of equivalent graphs with the best standard BIC score, amongst all the searches, is returned as the search output. On average, the genetic algorithm outperforms the PC algorithm (Spirtes, Glymour, Scheines, 2001) on errors of edge omission, commission, and orientation. This study describes a few methods of guiding search, maintaining diversity, and dealing with bias in the population. Many of the findings should apply to discrete Bayesian Network model selection and graph search more generally.

0. Introduction

Structural Equation Models are powerful tools used throughout statistics, economics, and the social sciences. SEMs facilitate efficient inferences from observations. They allow us to answer questions regarding predictions, interventions, and counterfactuals. The problem of searching for the SEM that best represents the observed data is difficult due to the quantity of directed acyclic graphs that exist for any number of variables. In this paper, we introduce three contributions to the current causal genetic algorithm literature, one of which is specific to SEMs; the other two apply to directed acyclic graph search. The SEM specific contribution is a genetic operator that considers trimming edges of low weight from the SEM. The second is a method of trimming the population that is similar to a crowding operator but that selects organisms for termination based on both how different they are from a defined set and their fitness. The third and most influential contribution is to keep multiple completely independent populations, each biased differently. The different biases have different levels of success on graphs of different complexities and won't necessarily fall victim to the same local maximum traps. The bias is primarily achieved by affecting the fitness function while remembering the graph that would have scored the best on the asymptotically correct Bayesian Information Criterion approximation. The secondary influence of the bias is achieved by setting the parameters randomly.

The structure of the paper is: First, we present a short background on SEMs, causal search, genetic search, and the goals one strives for in genetic algorithm development. Second, we describe the genomic representation that we use to represent DAGs and explain how the functions applied to organisms work. Third, we detail: how the main loop of the algorithm works, and how the population is created, regulated, and maintained. Fourth, we describe two heuristics used by the genetic algorithm. One is the SEM specific operator mentioned above. The other is a simple hill climb that explores the immediate local fitness space when the algorithm thinks it is finished. Fifth, we address the issue of bias and discuss search parameterization. Sixth, the effect of combining multiple randomly parameterized searches is discussed. Seventh, the genetic algorithm is compare to the PC algorithm on simulated data. Finally, we speculate on improving the performance of the algorithm in the future.

1. Background

Model Description (SEMs)

A structural equation model (SEM) is represented by a directed acyclic graph (DAG) in which the vertices are variables. The DAG expresses which variables are direct causes of a given response contained within the graph. A linear equation expresses each variable as a function of its direct causes and a "noise" term. Linear SEMs with variables expressed as a deviation from their mean use the simple function $Y_i = b^T X_i + \varepsilon_i$, where Y_i represents response variable Y's value at individual i, X_i represents a vector of Y's parents values at individual i, b represents a vector of the linear contributions to Y of each of Y's parents. . And ε_i represents the residual or error term. For a detailed explanation of DAGs, SEMs, and linear SEMs see (Pearl 2000, Bollen, 1989)).

Causal Search

The search space involved in DAG exploration is astronomical. For a given number of variables n, there are $\frac{n(n-1)}{2}$ possible adjacencies that can exist in a graph. Each of these adjacencies can be present in a given graph or not making the number of possible adjacency structures $= 2^{\frac{n(n-1)}{2}}$. Each of these adjacencies can then be directed as long as no cycles are generated in the graph. The size of the hypothesis space can be computed using a recurrence relation (Harary, p.19) that sits between $2^{\frac{n(n-1)}{2}}$ and $3^{\frac{n(n-1)}{2}}$. Over a set of 10 variables, 4.1751E18 unique DAGs can be constructed. In order to deal with this sizeable hypothesis space, two basic methods have been proposed: Constraint based search and Scoring based search. Constraint based search (Spirtes, Glymour, Scheines, 2001) uses independence relationships and conditional independence relationships inferred from observed data to determine the adjacencies and then through constraint propagation orient as much of the graph as possible. The main advantage constraint based search algorithms offer is their relative speed, their ability to handle latent variables, and the availability of asymptotic consistency proofs. Their two main draw backs are 1) that when transforming the data into a set of independence relations, an incorrectly assigned independence relation has the potential to propagate through additional independence relation calculations, and 2) in cases in which no DAG entails all and only the independence relations judged to hold in the data, the algorithm has no way to search for the DAG that sits "closest" to the impendence structure judged to hold. Scoring based search has nearly the complementary set of advantages and disadvantages. Because proximity among DAGS, at least with respect to small perturbations in adjacencies and/or orientation, translates terribly into scoring proximity over DAGs, local maxima abound and a search for the model or models with the best "score" is hard. As a result, scoring based searches are typically very slow.

Genetic Search

Genetic Algorithms, a subset of scoring algorithms, search for multiple solutions simultaneously. Over time these solutions are blended with each other and are maintained in a population based primarily on their fitness. The hope is that characteristics of the real model will improve fitness and be selected for over time. All genetic algorithms follow some sequence of decisions that can be transformed into the following format, to mimic natural selection.

1) Generate initial population
2) Select a subset of the organisms from the present population
3) Produce offspring from crossing different organisms
4) Mutate the current population
5) Return to 2

Diversity & Evolution

For a search to perform well, it must be able to adapt to the fitness landscape. A hill-climbing algorithm can easily solve a landscape in which the fitness function increases or decreases smoothly with adjacent graphs in the search, there are no local maxima (the surface is concave), and the fitness function favors the true model. Unfortunately, the fitness landscape of scoring functions of causal models is not concave. All we can realistically hope for is that the fitness landscape is concave around the real solution. If this is the case, all we need to do is find a solution that is reasonably close to the real model and let hill climbing finish the job for us once inside the concave region. Since the models explored by the algorithm are probabilistically determined by the current set of solutions, strong similarity amongst the organisms can produce organisms that do not differ significantly from the rest of the population. If the current population is situated in a concave region that does not contain the real solution, the search can become trapped in a local maximum. To avoid this precarious scenario, many methods of maintaining diversity in the population have recently been attempted. Crowding, explored by Cavicchio (1970), De Jong (1975), and Mahfoud (1992), is one such method where organisms involved in mating have the potential of being removed from the population. Crowding varies based on the specifics of what constitutes a reason for trimming a parent from the population. Multiple sub-populations, explored by Grosso (1985), represent several independent islands, upon each island, the GA runs as normal while migration between the islands is allowed but restricted. These theories derive from the fact that search success is dependent on two criteria that intuitively oppose each other. Diversity instructs us to favor selection of organisms given how different they are, while the fitness function will direct us towards selecting the best models, which have likely been biasing the population for many generations.

2. Representation & Operators

Genome

Each DAG is represented by a genome or sequence of traits, one for each possible edge between any two variables found in the graph. The alleles are represented by the values of the individual traits, each trait represents an edge's orientation or absence from the model. For each variable in the model, a singular value decomposition regression calculates a regression intercept and a linear coefficient for each of the variable's direct causes. Therefore, each genome/graph uniquely generates a SEM given the observed data.

Mutation

One or more random alleles in the genome are changed to a different random value: that is, an edge is reoriented, removed, or added to the current model. The number of alleles changed in a mutation is determined by the mutation rate. The mutation rate and best mutation rate are parameters of each search. The best mutation rate is a measure of how much to mutate copies of the best model (See Mating Cycle). The value to which an allele is changed is probabilistically determined by a mutation distribution or remove to reorient ratio. The standard distribution used if not declared otherwise is to remove an edge 2/3 of the time and orient with even probability the rest of the time. One must note that the mutation operator is not closed. If the mutated graph contains a cycle, it will be passed through the random cycle breaking operator.

Crossover

Given two organisms, a random mask of the same length as the genome is generated. Each organism maintains the alleles covered by the mask; the alleles not covered are swapped with the other organism. As with the mutation operator, crossover is not closed and will often need to pass its results through the random cycle breaking operator, described in the next subsection

Random Cycle Breaking

Given a cyclic graph, this procedure isolates cycles and randomly removes edges, contained in those cycles, until the graph is a DAG.

Scoring

The scoring function is a modification of the Bayesian Information Criterion (BIC), an approximation of the posterior probability of the SEM given the observed data. The basic BIC score can be broken into two parts: A measure of how similar the covariance matrix of the data is to the covariance matrix implied by the model at the ML parameter estimate, and a penalty based on how complex the model, which in the case here is equivalent to how many edges exist in the graph. Our modification to improve diversity is to treat the penalty contribution as a parameter of the search. The benefit of this modification is that searches can be biased toward edge commission or edge omission. The penalty ratio of a search is the main determinant of which graphs will be considered. Anytime an organism's rank in the population is discussed, rank is based on the weighted BIC approximation.

3. The Algorithm

Seeding

The initial population is seeded by an execution of the PC algorithm (Spirtes et al, 2001, p.84), a basic constraint based search. The PC search returns an equivalence class of DAGs, or pattern. The algorithm produces multiple graphs by randomly orienting all unoriented and bi-directed edges in the equivalence class graph. If any cycles are created by the random orientation, the graphs are passed through the random cycle breaking filter.

Mating Cycle

The Mating Cycle represents the main loop of the algorithm. In each mating cycle a litter of graphs is generated. The contents of the litter are produced by a probabilistic function of the present population given the parameters. In each mating cycle, the structural equation model genetic algorithm (SEMGA) will produce the litter from as many as three sources. The primary source is through crossover and mutation. Each mating cycle, the algorithm selects two random individuals from the population and mates them. Selection is based on ranking, the probability of an organism, g, being selected for mating in a given cycle is $\frac{\sqrt{rank(g)+1}-\sqrt{rank(g)}}{\sqrt{size(pop)}}$. The two selected individuals produce six solutions that are added to the current population. The six solutions are comprised of mutations of each of the original parents, the parents once they've been crossed with each other, and a mutation of a copy of each of the post crossover graphs. The secondary constituents of the litter are multiple mutations of the best solution. The function of mutating the best solution is to give the algorithm a chance to learn more about the solutions near the best graph seen. The higher the number of mutated copies of the best model added, the more the algorithm behaves like a simple hill climbing algorithm. The quantity of mutated copies produced each cycle is a parameter of search. Finally, the heuristics will add additional organisms to the litter when certain events happen such as when a new best graph is found or the best has stayed the best for x number of mating cycles. Once the litter is constructed, it is added to the population. As each of these new organisms is added, they expel the lowest scoring organism found in the population, assuming the population is at its maximum size.

Population

The Population dictates which elements are considered for selection at each mating cycle. Population size fluctuates on a 50 mating cycle loop. At the beginning of the loop, the population is given a maximum increase of 80 organisms. At the half loop, the population is trimmed in a way dictated by the trimming scheme. The basic cycle is similar to that of a natural world with seasons. In spring, living is easy while winter selects a subset of the population to die.

Trimming Scheme

Organisms differing by more then 300 points from the best model are removed from the population. The population is then decreased even further, by removing repetitious graphs from the given population. Organisms are trimmed based on how different they are from the set of best scoring organisms. Each step in the loop the graph with the least amount of difference from the set of best organisms and the lowest score is removed from the population. The set of best graphs grows each cycle, the best scoring graph not removed from the population and not already in the set is added. The score of how much an organism differs from the set of best graphs is the product of how much the organism differs from each individual in the set. The difference of two individuals is simply a count of how many edges the two graphs make different assignments for. This population scheme is meant to intentionally clean-up potentially stagnant environments. Many solutions very near each other disallow competing alleles from entering the population. Situations of stagnation are one of the main concerns associated with some of the considered heuristics. Methods aimed at maintaining diversity have recently been popular in genetic search over Bayesian Networks and Decision Trees. Trimming of this genre can be compared to natural behavior of animals such as avoiding incest and competing violently with other organisms of the same species.

Termination

Search generally terminates after running for 1000 mating cycles without finding a new best model. The termination criterion can be affected by some of the heuristics. Regardless of whether graph scores are still increasing, search stops if the number of mating cycles reaches 25,000.

Result Frontier

The results of each search are the set of all graphs that scored better than all others for any possible positive range of the penalty factor. This frontier of organisms should range from the simplest to most complicated reasonable explanations of the data. Each of these graphs is then reduced to the Markov equivalence class to which it belongs. Records of the entire frontier are retained allowing the search to know how each explored graph would rank using any penalty ratio.

4. Extra-Evolutionary Characteristics (Heuristics)

Fat Trimming

Fat Trimming is a heuristic that adds additional organisms to the population every time a new best model is found. The set of models added are a subset of all the graphs one can construct by removing a single edge from the best model. A graph is in the subset if and only if a 90% confidence interval of the removed edge's coefficient contains 0. The trimming is meant to descend upon the simplest model that sufficiently represents the data. The main fault to the fat trimming algorithm is that it can pollute the population with many graphs that are very similar to the new best.

Aggressive Hill Climbing

Aggressive Hill Climbing is a last ditch chance to make sure that no superior graphs can be found locally. Aggressive Hill Climbing inserts every possible single edge mutation of the best scoring graph. The heuristic generates num variables * (number of variables –1) graphs, heavily polluting the population. For this reason Aggressive Hill Climbing is executed once a search would normally terminate. If a better graph is found search continues normally.

5. Parameter Choice, Resulting Bias

Parameter Choice

Parameter choice is a complicated problem. Although the parameters exert a definite influence upon the search, this influence is often hard to isolate due to the probabilistic nature of the search. For this reason, optimal values for the search parameters are impossible to calculate with any measure of confidence. Although we can't know the best way to set the parameters, we can identify certain values of the parameters as fruitless. According to previous analysis[1], the acceptable ranges for the parameters are given here:

Mutation Rates $\in [1,4]$
Lead Copies $\in [1,3]$
Penalty Ratio $\in [0.8,2.6]$
Any combination of the heuristics

Bias

Each parameterization biases the graphs explored by the search. Unfortunately, the magnitude and direction of the bias is often hard or impossible to determine for specific parameters. Mutation rates bias search if the mutation distribution doesn't equal the real model's edge density. The penalty ratio used in the modified BIC score is the only variable whose bias upon the sample can be assumed. The larger the penalty ratio, the simpler the set of models considered for scoring and vice-versa. Besides the penalty ratio, the current population is the largest bias of the search. The only way for an allele not found in any organism in the population to enter the population is through mutation. If the score of the graph can't be improved within a certain number of mutations, search falls into a local maximum. The propensity for an algorithm to fall into a local maximum is dictated by the mutation rates, the number of copies made of the best each mating cycle, and the heuristics.

6.The Master Search

The term search has been used to describe a single execution of the SEMGA. The actual algorithm, called the master search, executes multiple searches with random acceptable parameterizations. From each of the searches, the model that scores the best using the standard BIC penalty ratio is chosen as the model selected by the search. If we had an idea of how complicated the real model was, we could tailor a search that would be more likely then a random search to find the real model. We cannot know the real model's complexity but we can try multiple parameterizations, each resulting in its own search bias. Hopefully, one of the parameterizations will produce the proper bias. In the minimum, this search structure maintains multiple independent populations of evolving graphs. If one search stagnates at a local maximum, the other searches may avoid it.

7. Results

Scoring Graph Fitness

The equivalence class returned by the master search is compared to the equivalence class to which the real model belongs. Equivalence classes are used because all graphs in an equivalence class should score the same minus a little numeric instability from the computer. Any potential

[1] The ranges represent the threshold at which we observe a noticeable degradation in search performance over 3-35 variable models.

edge between two graphs can differ in four ways: (let x denote the graph being compared to the original)

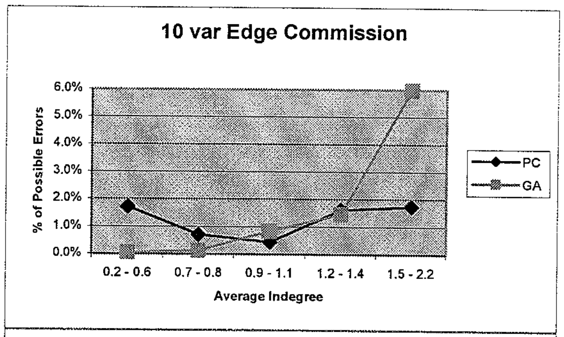
- Edge commission: edge not in equivalence class of real model but found in x
- Edge omission: edge found in equivalence class of real but not in x
- Orientation commission: edge found in both but x orients the edge differently
- Orientation omission: edge found in both but x doesn't choose a direction while the real model has enough information to orient the edge

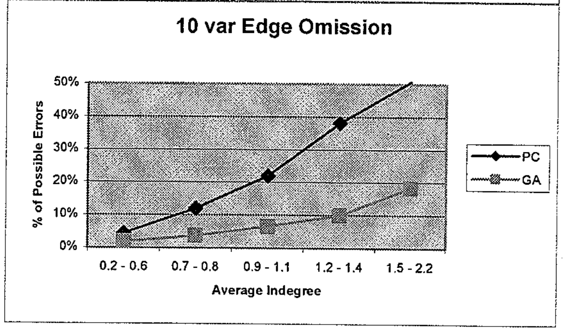
The results are returned as a % of possible errors. The quantity of possible edge commissions is equal to the number of potential edges in the graph minus the number of edges in the real model. The number of possible edge omissions is equal to the number of edges in the real model. The number of possible orientation commission and omission errors is equal to the number of edges that the real model and best model have in common.

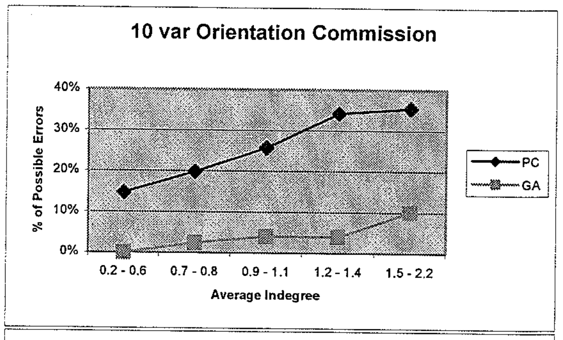
Simulation

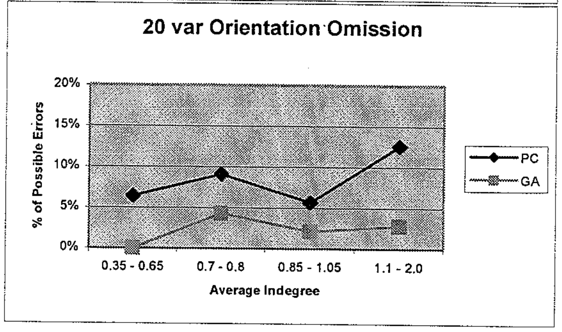
The following graph represents the results of search on 248 simulated data files. Each of the data files consists of 100 individual values of 10 variables produced from the same randomly generated SEM. The searches have been divided into 5 groups based on average indegree in order to achieve reliable results upon averaging. The breakdown of the groups is:

53 models with indegree ranging from 0.2-0.6
45 models with indegree ranging from 0.7-0.8
58 models with indegree ranging from 0.9-1.1
39 models with indegree ranging from 1.2-1.4
53 models with indegree ranging from 1.5–2.2



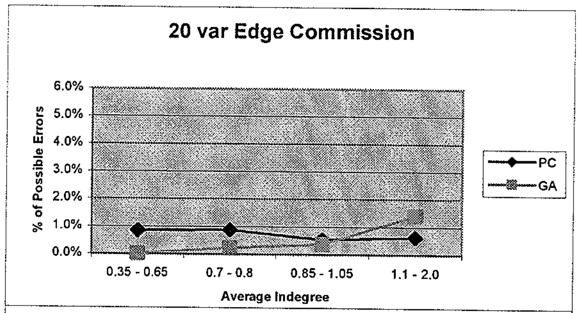


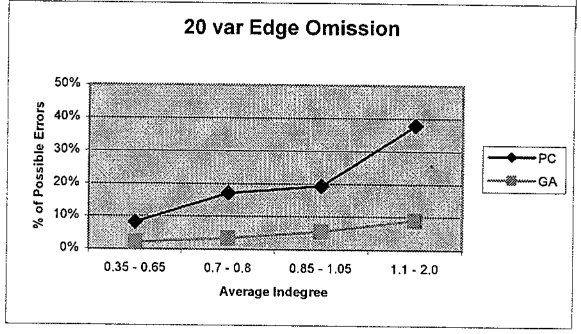


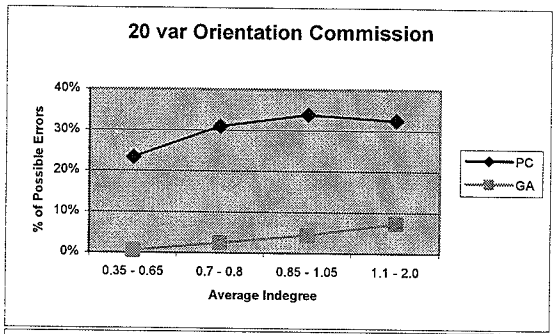


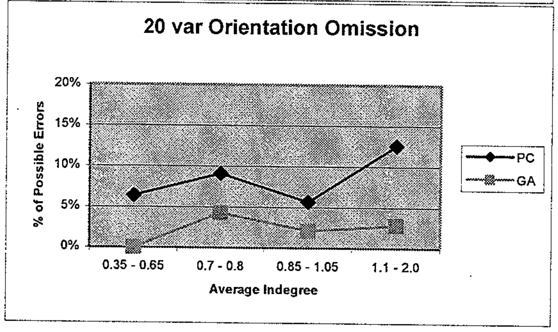
This next set of graphs represent the results of search ran on 138 simulated data files. Each of the data files consists of 100 individual values of 20 variables produced from the same randomly generated SEM. The searches have been divided into 4 groups based on average indegree in order to achieve reliable results upon averaging. The breakdown of the groups is:

31 models with indegree ranging from 0.35-0.65
34 models with indegree ranging from 0.7-0.8
39 models with indegree ranging from 0.85-1.05
34 models with indegree ranging from 1.1-2.0









The results provide strong evidence of the SEMGA almost dominating the PC algorithm on data generated by linear structural equation models with no latent variables. The SEMGA produces fewer errors on average with the exception edge commission, which is quite small for both the PC and Genetic Algorithms.

BIC considerations

The score of the true graph model's relation to the frontier is important. The range in which the true model dominates all models explored defines a region where a local maximum was found and search stopped. Local maximum graphs are solutions upon which search might have done better, had it been allowed more time or diversity control. The range of penalty ratio values where the real model is dominated by some other model defines a range where the solution will not asymptotically converge to the correct solution.

9. Future Work

The plans for the SEMGA's future are threefold: First, the search should extended to cyclic graphs and graphs that contain latent variables. Second, new genomic representations are being considered. There is a chance that a representation of a causal ordering mapped over an adjacency structure may encode the data better, allowing for more intelligent crossover and mutation. This representation has been used frequently in Bayesian Network structure learning (deCampos). Third, the algorithm should evolve over time. For example the density of the mutation operator could reflect the solutions we've already seen. The frontier of best solutions contains reasonable estimates of the hardest to break dependence relations and most complicated generalizations the algorithm might need to explore. These features may offer insights into a better mutation function density or may be able to reduce the size of the search space by assuming certain traits as truths and others as impossibilities.

References

1) Spirtes, P., Glymour, C., & Scheines, R. (1993). *Causation, prediction, and search.* SpringerVerlag, Berlin.

2) Bollen, K.A. (1989), ``Structural Equations with Latent Variables", New York: John Wiley and Sons.

3) DeJong, K., (1975), *An Analysis of the Behavior of a Class of Genetic Adaptive Systems.* Doctoral dissertation, University of Michigan.

4) Mahfoud, S. (1992). Crowding and preselection revisited. *Parallel Problem Solving from*

5) Cavicchio, Jr., D. J. (1970). *Adaptive search using simulated evolution.* Doctoral dissertation, University of Michigan, Ann Arbor, MI (University Microfilms No. 25-0199).

6) Grosso, P.B. (1985). *Computer simulations of genetic adaptation: Parallel subcomponent interaction in a multilocus model.* Unpublished doctoral dissertation, The University of Michigan. (University Microfilms No. 8520908)

7) deCampos, Jose A., Gamez, Serafin Moral (1999). *Computation Approach by using Problem Specific Genetic Operators.* University of Spain, Grenada

8) Harary, Frank (1973). *Graphical Enumeration.* Academic press, New York and London

9) Pearl, Judea. (2000). *CAUSALITY: Models, Reasoning, and Inference*
(Cambridge University Press, 2000)