Proceedings of Student/Faculty Research Day, CSIS, Pace University, May 4th, 2007

EXPERIMENTATION AND DISCOVERY IN GRAPH THEORY USING AN EVOLUTIONARY TOOL

Michael L. Gargano, mgargano@pace.edu
Computer Science, Pace University, NYC, NY 10038

Lorraine Lurie, llurie@pace.edu
Mathematics, Pace University, NYC, NY 10038

ABSTRACT

A general problem confronting graph theorists working on problems that are NP-hard is finding good examples so that reasonable conjectures can be produced. We propose an order based hybrid evolutionary modified life cycle model to help in experimenting and discovering patterns in some types of graph theory problems involving weight related labelings.
Keywords: labelings, weights, graph theory

EXAMPLES OF SOME PROBLEMS

The minimum sum vertex cover problem

Let G be a graph with vertex set V(G) and edge set E(G). A vertex labeling is a bijection f from V(G) to $\{$ 1, 2, …, card (V(G)) $\}$. The weight of an edge e = { u, v } in E(G) is given by f(e) = min { f(u), f(v) }. The minimum sum vertex cover (**msvc**) problem seeks a labeling that minimizes the sum of all the edge weights in G.

The neighborhood sum problem

Let G be a graph with vertex set V(G) and edge set E(G). A vertex labeling is a bijection f from V(G) to $\{$ 1, 2, …, card (V(G)) $\}$. The neighborhood sum (**ns**) problem seeks a labeling that minimizes the sum of all the vertices in any closed neighborhood of G.

There are many other similar optimal labeling problems in graph theory.

We propose an order based hybrid evolutionary algorithm, a biologically inspired heuristic search procedure, to provide experimental results for these problems and thus aid in developing conjectures. This algorithm primarily identifies an optimal, and/or near optimal, permutation labeling (bijection) in a computationally efficient manner.

The life cycle paradigm [4, 9] is an adaptive method evolving a population of potential solutions into a new, fitter, population of potential solutions using swarm methods, genetic algorithms, and local search. The process repeats itself until it reaches an optimal (or near optimal) solution.

HYBRID EVOLUTIONARY METHODOLOGY

We propose an order based hybrid evolutionary algorithm for these types of graph theory labeling problems. The implementation is based on prior research done in other application areas [4, 9]. Our method creates and evolves a population of potential solutions (i.e., permutations (or bijections from V(G) to $\{$ 1, 2, …, card (V(G)) $\}$) so as to facilitate the creation of new members by swarming [4, 9], mating and mutating [1, 2, 3, 4, 5, 6, 7, 9], or local search [4, 9].

Fitness (or worth) is naturally scored by the minimum sum of all the edge weights in G for the minimum sum vertex cover (**msvc**) problem and the minimum sum of all the vertices in any closed neighborhood of G for the neighborhood sum (**ns**) problem.

This algorithmic method is a hybrid combining swarm methods, genetic algorithms, and local search. An individual population member passes through three phases that are iterated until a satisfactory solution is obtained. As in many processes in nature, each individual member goes through different life cycle paradigms as it evolves. In this adaptive search heuristic a member goes from a swarm search to a genetic search to a local search and back again reiteratively.

MODIFIED LIFE CYCLE MODEL

We now state the modified hybrid life cycle algorithm that we used:

```
 Randomly initialize a population of possible solutions in each paradigm.
 For all swarm members
         Swarm
 For all genetic algorithm members
         Mate and Mutate
 For all local search members
         Search Locally
While (terminating condition not met)
         For each member
                 Switch life cycle phase if no recent improvement
         For all swarm members
                 Swarm
         For all genetic algorithm members
                 Mate and Mutate
         For all local search members
                 Search Locally
```

Since each phase has difference characteristics, this hybrid paradigm utilizes the best of each. Some of the differences are follows:

Swarm	**Genetic Algorithm**	**Local search**
Movement	Replacement	Movement
Directed Change	Random Change	Greedy Change
Self-organizing	Natural Selection	Greedy
Cultural Transmission	Recombination	None
Neighborhood	Non-local	Neighborhood

SWARM PHASE

In this phase each swarm population member p_i swarm through permutation space. The global best of all the members is noted as member p_g. After being influenced (via cultural transmission) by the global best p_g, a particle (member) searches locally by considering all of its neighbors (i.e., permutations that results from a simple transposition of two adjacent positions in the permutation, also known as a simple 2 reversal). Each member then moves to a permutation neighbor of best fitness or stays in place.

Here is the algorithm:

Randomly initialize a swarm population of permutations in permutation space
Repeat
    Find the global best p_g
    For each member of the swarm
        Mimic part of the global best p_g and move there
        Consider all of the neighbors and move to a neighbor of best fitness or stay in place

For example, if $g = 6$ and member p_g = 5 2 6 4 1 3 is best then member p_i = 3 5 2 4 6 1 may first mimic member p_g by moving to 2 5 3 4 6 1 since 2 comes before 3 in p_g, then search locally by considering all of its neighbors (i.e., permutations that results from a simple transposition of two adjacent positions in the permutation, also known as a simple 2 reversal). Thus, for example a neighbor would be 5 2 3 4 6 1 by reversing the first two positions.

If a member has no place to move or it has the same fitness for some time, then this member will be removed from this population and be put into the genetic algorithm population.

GENETIC ALGORITHM PHASE

The mating convention is such that only high scoring fitness members will preserve and propagate their "worthy" characteristics from generation to generation and thereby help in continuing the search for an optimal solution. GA implementation requires a suitable encoding in chromosome space to evolve the chromosome members of the population and a mapping of the chromosome into permutation space to score the permutation members of the population. The permutation representation used is related to the Cantor Expansion of an integer [8] and the methodology which we use is similar to that used in the GA solution of the Traveling Salesperson Problem [2]. Each member of the population of potential solutions is encoded as a sequence of g genes in a chromosome where g is equal to card (V(G)). The value j of the gene in the k-th position of the chromosome (i.e., v_k) can range between 1 and k where chromosome position is measured from right to left starting at position 1. Thus, the chromosome can be represented by $v_g, v_{g-1}, \ldots, v_l$ where $v_k = j$ $(1 < j < k)$. This chromosome encoding is mapped into a g-th order permutation as follows: For the chromosome position index k ranging from g down to 1, place k at position i of the permutation (i.e., P_i). That is $P_i = k$ where $i = c(j)$. Here c is a counting function that determines i as the j-th unfilled position in the permutation counting from right-to-left. A simple example illustrates the above. If $g = 6$, a chromosome encoding obeying the range bounds for each position is: (452221) and its corresponding permutation is: (526341).

Applying this permutation to the vertices of G yields a labeling of V(G) from which the fitness can be scored.

Selection of parents for mating involves choosing one chromosome member from the high scorers and choosing the other chromosome member randomly. The reproductive process is a simple crossover operation where two selected parent members are cut into segments at some randomly chosen positions and then have their segments swapped to create two offspring members. The crossover operation yields offspring chromosomes whose genes always satisfy the range bounds. A grim reaper mechanism replaces low scoring members in the population with newly created higher scoring offspring. Mutation is a GA mechanism where we randomly choose a chromosome member of the population and change a few randomly chosen genes of that chromosome. This process is useful in creating new areas of search to avoid getting caught on local minima of the solution space.

We can now state the genetic algorithm that we used:

Step 1: Randomly initialize a population of chromosomes (genomes).
Step 2: Map chromosome members to permutation members (phenomes).
Step 3: Score any member that has not yet been evaluated.
Step 4: Sort the members of the population by their scores.
Step 5: Select one parent for mating from the upper portion (fitter) of the population and the other one randomly.
Step 6: Generate offspring using simple crossover.
Step 7: Mutate randomly selected members of the population at randomly selected genes at each generation.
Step 8: Replace the lower half of the population with offspring.
Step 9: If fitness has not changed recently
or a certain prespecified number of iterations have been made
or a known lower bound has been reached or nearly reached
Then return solution(s) found
Else go to Step 2.

If a member has the same fitness for some time or a certain prespecified number of iterations have been made, then this member will be removed from this population and be put into the local search population.

LOCAL SEARCH PHASE

In this phase each member considers all of its neighbors (i.e., a permutation that results from a simple transposition of two adjacent positions of the permutation, also known as a simple 2 reversal). Each member then moves to a permutation neighbor of best fitness (if there are two or more choose one randomly).

If a member has no place to move or it has the same fitness for some time, then this member will be removed and be put back into the swarm population.

HYBRID EVOLUTIONARY METHOD ADVANTAGES

The advantages of this approach are:

1). it will usually produce an optimal and/or a near optimal solution(s).
2). convergence to the optimal permutation can be established when just one member of the population scores a minimum (or maximum) which is known apriori
3). its computational complexity is polynomial.
4). it can easily handle constraints and incorporate heuristics and/or best known results.
5). it is faster and more effective than any of swarm, genetic algorithm, or local search individually

CONCLUSIONS

We considered using a hybrid evolutionary model based on a modified life cycle paradigm in order to experiment and discover in graph theory problems involving weight related labelings when the problems are NP-hard. We then argued that using this self-adapting algorithm that employs various different properties of the well known search strategies improves the algorithm's efficiency. In solving this NP-hard problem, the algorithm employed different life cycle phases when appropriate and adapted to its current search needs making it more efficient.

ACKNOWLEDGEMENTS

We wish to thank Pace University's Seidenberg School of Computer Science and Information Systems for partially supporting this research.

REFERENCES

[1] Dewdney, A.K. 1988. The Armchair Universe - An Exploration of Computer Worlds, W. H. Freeman & Co. pp 252.

[2] Davis, L., 1991. Handbook of Genetic Algorithms, Van Nostrand Reinhold.

[3] Edelson, W. and Gargano, M. L. 1995. A Genetic Algorithm Approach To Optimizing Portfolio Merging Problems. The Proceedings of The Third International Conference on Artificial Intelligence Applications on Wall Street. pp 168-173.

[4] Gargano, M.L. and Lurie, L. 2006. A Hybrid Evolutionary Approach To Solving The Archaeological Seriation Problem, Congressus Numerantium, 180, pp. 43-54.

[5] Gargano, M.L. and Rajpal, N. 1994. Using Genetic Algorithm Optimization To Evolve Popular Modern Abstract Art. Proceedings of the 1994 Long Island Conference on Artificial Intelligence and Computer Graphics. pp. 38-52.

[6] Gargano, M. L. and Edelson, W. 1996 A Genetic Algorithm Approach To Solving The Archaeological Seriation Problem, Congressus Numerantium, 119, pp. 193-203.

[7] Goldberg, D., 1989. Genetic Algorithms, Addison Wesley Publishing Co.

[8] Rosen, K. H. 1991. Discrete Mathematics and Its Applications. Second Edition. McGraw-Hill, pp 129, 286.

[9] Uran, B. and Gargano, M. L., 2005. Discovering Effective, Comprehensible Data Classification Rules Using Hybrid Evolutionary/Swarm Model for Data Mining, Congressus Numerantium, 176, pp. 49-63.