A Modified Genetic Algorithm for Matching Building Sets

Andrew Buck, Isaac Sledge, James Keller, and Marjorie Skubic
Computational Intelligence Research Lab – University of Missouri

Abstract

In this project, we develop a technique to search for a set of buildings in a geospatial database based on spatial relationships. We use a modified genetic algorithm to create a population of potential building sets which evolve toward a best solution. Each building in the database is spatially related to its neighbors through the histograms of forces. Our modification to the mutation operation of the genetic algorithm allows us to direct candidate solutions towards the best match using the histograms of forces.

Background

In [1], Engelbrecht presents an overview of genetic algorithms (GA), which are modeled after natural evolution using genotypes. Each candidate solution is expressed as a chromosome containing a set of genes, one for each variable to be optimized. A fitness function evaluates how well each chromosome matches the ideal solution. Evolution occurs through the selection, crossover, and mutation operators. The selection operator uses the fitness values to pick a set of chromosomes for reproduction or survival to the next generation. Crossover creates offspring chromosomes by combining genetic material from a set of parents. Mutation randomly alters the genes of a single chromosome to introduce new genetic material. Traditionally a GA uses all of these operators on a population of chromosomes represented by a bitstring, where each bit represents a gene. In this project, we modify the traditional GA to work around the limitations of our problem domain. To understand the details of our modification, we must first look at how the building database is structured. Our experiments are run on an area of downtown Columbia, MO containing 2467 buildings. Each building is connected to its 30 nearest neighbors by their histograms of forces (HoF). Given two buildings, A and B, $F^{AB}(\theta)$ is the histogram of forces from A to B and represents the support for which A is in direction θ of B [2]. The database does not store any absolute position data about the buildings, only their relative directions.

Algorithm

Each chromosome in the GA is a set of potential buildings. Each building is represented by a gene in the chromosome. The algorithm starts by creating a random population of chromosomes over the search space with the restriction that each set must be fully connected, meaning that there is a HoF from every building in the chromosome to every other building in the chromosome. Ideally these chromosomes will be evenly distributed over all the buildings in the database. We create new chromosomes for the next generation by applying the mutation operator to the entire population. Because each chromosome represents a set of buildings that are close together, the combination of sets through crossover could result in child chromosomes that are not fully connected, so mutation is used exclusively. The mutation operator begins by randomly selecting one of the genes to replace. This gene corresponds to one of the buildings in the target set. The histograms relating this building to all the other buildings in the target set are stored as the target histograms. The set of nearest neighbors is the set of buildings which could replace the chosen gene and maintain full connectivity. Each nearest neighbor is evaluated against the target histograms to see how well it would "fit" into this spot. The building with the best fit is chosen as the new value for the gene.

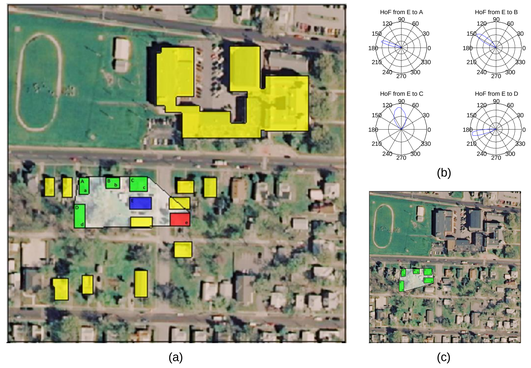


Figure 1: The process of the mutation operator just before convergence (a) and after (c). The chromosome contains buildings a-e (four green and one red). The target set contains buildings A-E (four green and one blue). Building e in red is randomly chosen for replacement. (b) shows the histograms of forces relating E to buildings A-D. The nearest neighbor (shown in yellow except for E in blue) that most closely matches these histograms is chosen to replace building e in red. Because this is the last mutation before convergence, building E in blue is a perfect match. (c) shows the final chromosome, perfectly matched with the target set.

Histograms are compared using their cross-correlation, which determines the amount that they overlap. This can be calculated using the formula:

$$\mu_c(h_1, h_2) = \frac{\sum_\theta h_1(\theta) h_2(\theta)}{\sqrt{\sum_\theta h_1^2(\theta) h_2^2(\theta)}}$$

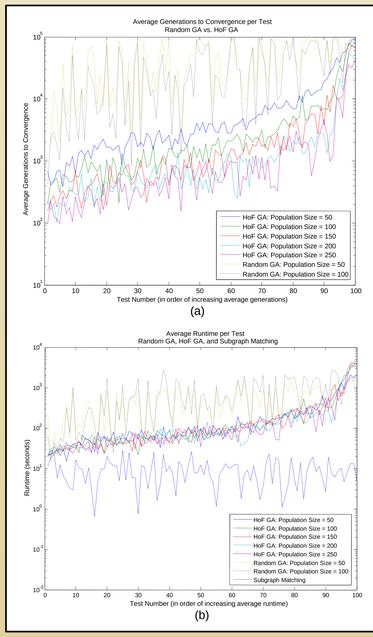
We evaluate the fitness of a chromosome by comparing all of the histograms of the search set with the histograms of the target set. When a chromosome is mutated, its fitness determines if it will replace its parent by the formula,

$$P(x_c(t)\ wins) = \frac{f(x_c(t))}{f(x_c(t)) + f\left(x_p(t)\right)}$$

Where x_c is the child chromosome, x_p is the parent chromosome, and $f(x_i(t))$ is the fitness of chromosome x_i. A parent with a lower fitness value than its child is more likely to be replaced. This helps prevent a low-fitness child from replacing a high-fitness parent. The algorithm continues until a perfect match is found or the generation limit is reached.

Results

To test our modification to the traditional GA, we compare it to a GA that ignores the HoF relations and simply mutates a chromosome by replacing a random building with one of its nearest neighbors. We use 100 test sets of five buildings and evaluate the average performance of each algorithm over ten runs. We use population sizes of 50, 100, 150, 200, and 250 for the HoF GA, and population sizes of 50 and 100 for the random GA. We compare the computational runtime and the number of generations to convergence with a maximum of 100,000 generations. The HoF GA has a much higher convergence rate than the random GA and typically requires roughly ten times fewer generations to converge. Larger population sizes tend to result in fewer generations, although there are some exceptions. The HoF GA also has a faster runtime than the random GA, but is not dependant on the population size. This is due to the greater computational complexity of the larger populations. Although the HoF GA performs better than the random GA, both GA types have slower runtimes when compared against a nondeterministic subgraph matching algorithm.



Percentage of Tests That Converge Within 100,000 Generations		
GA Type	Population Size	Percent of Convergence
Random GA	50	69.4%
	100	79.5%
HoF GA	50	96.9%
	100	98.0%
	150	98.9%
	200	99.4%
	250	99.6%

(c)

Figure 2: (a) Average generations to convergence per test, comparing the random GA to the HoF GA. The HoF GA requires fewer generations on average than the random GA. Larger Populations result in fewer generations.
(b) Average runtime per test, comparing the random GA, HoF GA, and subgraph matching methods. The HoF GA executes faster than the random GA, but not as fast as the subgraph matching method. Population size has little effect on the average runtime.
(c) Percentage of tests that converge within 100,000 generations. The HoF GA has higher convergence rates than the random GA, and large population sizes have a greater probability of converging within the generation limit.

Conclusion

In this project, we developed a modification to the mutation operation of a GA for the purpose of building matching. This new operator uses the histograms of forces to select an ideal replacement building. Although the HoF GA does not perform as well as the subgraph matching algorithm, it does much better than a simple random GA. This demonstrates the usefulness of using HoF relations when comparing spatial relationships.

Acknowledgements & References

This work is funded by the U.S. National Geospatial-Intelligence Agency NURI grant HM 1582-08-1-0020.

[1] A. P. Engelbrecht, *Computational Intelligence: An introduction*, 2nd ed. Chichester, West Sussex: John Wiley, 2007, pp. 125-176.

[2] P. Matsakis, J. M. Keller, O. Sjahputera, and J. Marjamaa, "The Use of Force Histograms for Affine-Invariant Relative Position Description," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 1-18, Jan. 2004.

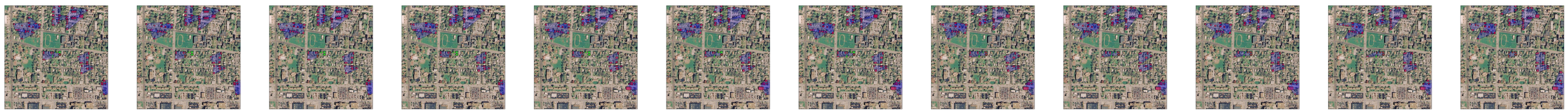


Figure 3: From left to right, an animation of the HoF GA as a chromosome converges on the target set in green. Each chromosome is represented by five red buildings with a blue overlay around the convex hull.