# Evolutionary Multiobjective Optimization for School Redistricting


## James Ryan Carr

Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County
1000 Hilltop Circle, Baltimore MD 21250
carr2@umbc.edu



## Abstract

I present an application of the Simple Evolutionary Algorithm for Multiobjective Optimization (SEAMO) to the problem of school redistricting. The application domain, evolutionary algorithms in general, and SEAMO in particular are outlined, the application of SEAMO to school redistricting is explained, and its performance is compared to other heuristic local search methods previously implemented in this domain. SEAMO is found to be comparable to these algorithms with respect to plan quality and diversity, but has the advantage that it optimizes all objectives simultaneously, instead of improving some at the expense of others.


## Introduction

This paper presents an application of evolutionary multiobjective optimization to the problem of school redistricting. It builds on the previous work of desJardins et al. [3], which presents a system designed to help school officials in Howard County, Maryland find and visualize possible solutions to this problem. This paper presents the results of implementing the Simple Evolutionary Algorithm for Multiobjective Optimization (SEAMO, developed by Mumford-Valenzuela [4]) in the system, and compares them to the results generated by the heuristic search methods already in the system.

## Background

This section outlines the application domain as presented by desJardins et al. [3], gives a general description of evolutionary multiobjective optimization, and presents the SEAMO algorithm as described by Mumford-Valenzuela [4].

### Application Domain

The goal of school redistricting is to assign each neighborhood (or *planning polygon*) in a county or school district to a school at the elementary, middle, and/or high school level. An assignment of each neighborhood in the school district to a school at one of these levels is called a *partition* or *plan*. There are many different criteria that school officials consider when creating plans. For instance, it is often desirable to minimize busing costs, meet school capacities, to balance distributions of socioeconomic status and test scores, and to allow students who live within walking distance of a school to attend that school. Improving one of these criteria often comes at the expense of others; for instance, in order to balance socioeconomic distribution, it may be necessary to bus some students in a wealthy neighborhood near one school to a school further away, which increases busing costs. Because of this, school redistricting is a multiobjective optimization problem. Therefore, there are a large number of possible plans that represent good solutions in the search space. In addition to finding good plans, it is often desirable to find plans that represent tradeoffs among the criteria (such plans are said to be *qualitatively different*) [3].

**Evolutionary Multiobjective Optimization**

Evolutionary algorithms (EAs) are a popular method for local search over a single objective, and they have been shown to quickly converge to the optimal solution [sources from 3]. However, when there is more than one objective (particularly when objectives are contradictory to one another), the search space is not well defined, so a new technique for searching with EAs is necessary. Evolutionary search over multiple objectives is known as evolutionary multiobjective optimization (EMO), and there is a relatively small body of work on EMO compared to single-objective evolutionary optimization [2].

Both single-objective EAs and EMO algorithms use natural selection as a method of exploring their search space for good solutions. Given an initial set of possible solutions (the first *generation*), the processes of selection, crossover, mutation, and replacement are applied to produce a new generation. Selection involves choosing a number of solutions (two, for example) to function as *parents* for a new solution. Crossover involves combining the attributes of the parent solutions in some way to create a *child* solution. Once the child solution has been created, it is subject to the process of mutation, which randomly alters at least one of its features. Finally, the process of replacement is used to determine which of the parent and child solutions are best, and to use them to populate a new generation. The aim is for each generation to contain solutions that are better than the generation before it. Thus, by allowing the algorithm to continue for a large number of generations, it should be possible to produce good solutions.

The primary obstacle for moving from single-objective EAs to EMO algorithms is implementing the selection process. Since there are multiple objectives, it is common for one solution to be better than another on some objectives and worse on others. EMO algorithms generally use one of three methods to deal with this problem: aggregating functions, population-based approaches, and Pareto-based approaches [2].

Aggregating functions deals with the problem by simply multiplying each objective by some weight and summing these values together; a higher sum indicates a better solution. Although clever applications of this technique have been shown to produce good results, there is little interest in new algorithms of this type [2].

Population-based approaches involve dividing the population into sub-populations based on the number of objectives the problem has. Therefore, a population with $M$ members in a problem with $k$ objectives would be divided into $k$ subpopulations, each with size $M/k$. Selection, crossover, and mutation are then run on each subpopulation, with only one objective used in the selection process for each. The subpopulations are then shuffled back together to obtain the new generation. This was the method used by some of the first EMO algorithms, and continues to be used in certain areas [2].

Pareto-based approaches incorporate the concept of Pareto optimality. A solution is said to *dominate* another solution if it is better on at least one objective, and is not worse on any other objective. A *Pareto-optimal* (or *non-dominated*) solution is one that is not dominated by any other solution [1]. Pareto-based approaches to EMO incorporate the concepts of dominance and non-dominance into the selection process in some way. These approaches have become the most popular methods for dealing with multiple objectives in EAs [2].

### SEAMO

The Simple Evolutionary Algorithm for Multiobjective Optimization (SEAMO) is a Pareto-based approach to EMO first presented by Mumford-Valenzuela [4]. Many EMO algorithms make use of fitness sharing, population-wide Pareto-dominance calculations, or other global calculations that are computationally expensive. SEAMO attempts to do away with these global calculations while still producing solutions that are both close to the Pareto front and widely and evenly spread along it.

In SEAMO, these objectives are reached primarily through the replacement strategy, which follows three rules. First, parents are typically replaced only by their own offspring. Second, offspring are only allowed to replace their parents if the offspring are superior. Finally, duplicates in the population are deleted. The first and third rules are used to ensure that the population remains diverse and does not converge to only one solution. This ensures that solutions are spread widely over the Pareto front. The second rule guarantees that the population can only improve, which guarantees that, given enough time, the algorithm will generate solutions close to the Pareto front [4].

SEAMO begins with an initial population of randomly generated members of the search space. In each generation, it performs selection by iterating through the members of the population. For each member, another member is randomly selected, and these two solutions are used as parents in a crossover to create one child solution. The child then has one mutation applied to it, and is then compared with its parents. If the child dominates one of its parents, it replaces that parent; if the child dominates both of its parents, it randomly decides which parent to replace; if the child does not dominate either of its parents, it is discarded, unless it contains the best value seen so far on any objective. In this case, the child is preserved by replacing one of its parents or, in rare cases, a different member of the population (this is an attempt to make sure that solutions are spread as widely as possible along the Pareto front). Once SEAMO has run for a certain number of generations (or some other stopping condition is reached), it returns the entire set of non-dominated solutions in the population [4].

SEAMO has been shown to outperform many popular EMO algorithms in some cases, and its performance relative to other algorithms has been shown to improve as the size of the search space of a problem increases [4].

## Methods

This section describes the evaluation criteria used in the search, gives a justification for choosing SEAMO over other algorithms for this application, and outlines the adaptation of SEAMO to the problem of school redistricting. Finally, it briefly describes the search methods previously implemented in the school redistricting project by desJardins et al [3].

**Evaluation Criteria**

Each generated school plan is evaluated along five dimensions chosen to represent the qualities that school officials look for when choosing plans. These functions are school capacity ($f_1$), socioeconomic distribution ($f_2$), test score distribution ($f_3$), busing cost ($f_4$), and walk area usage ($f_5$). Each function has been normalized so that its results fall in the range [0,1], with 0 being the ideal value. Note that these are the same evaluation criteria used by desJardins et al., and a more rigorous mathematical definition of each criteria may be found in their work [4].

**School Capacity ($f_1$):** Plans that have schools that are either significantly under- or over-utilized (i.e., the ratio of proposed enrollment to intended enrollment is below 90% or above 110%) are both considered poor. Thus, $f_1$ is a function that calculates these ratios and assigns a high value if a ratio is outside the range [0.9, 1.1].

**Socioeconomic Distribution ($f_2$):** The percentage of students who receive free and reduced meals (FARM) is used as an indicator of a school's socioeconomic distribution. Ideally, every school in the plan would have the same FARM percentage. Thus, $f_2$ is calculated by measuring the difference between each school's FARM percentage and the FARM percentage over the entire county. Each school's value is fed into a penalty function, weighted by the school's population, and combined to get the value for the entire plan.

**Test Score Distribution ($f_3$):** Maryland State Assessment (MSA) percentages measure the proportion of students at a school who achieve above a certain level on assessment tests. As with FARM percentages, the ideal plan would have the same MSA percentages for each school in the county. Thus, $f_3$ is computed the same way as $f_2$, with the difference between each school's MSA percentage and the county-wide MSA percentage fed into a penalty function and weighted based on the school's population.

**Busing Cost ($f_4$):** Busing costs are minimized if every neighborhood is bused to whichever school is closest to it. Thus, $f_4$ is computed by calculating, for each neighborhood, the difference between the distance to its assigned school and the distance to its closest school. This value is normalized by the distance to each neighborhood fourth closest school, since it is almost never reasonable to bus students farther than that. Each polygon's value is weighted by its population and combined into a single value for the entire plan.

**Walk Area Usage ($f_5$):** When students are within walking distance of a school, it is preferable to send them to that school. This is represented in $f_5$ by calculating the number of students assigned to a school within walking distance and dividing it by the number of students within walking distance of any school, regardless of assignment. This value is then subtracted from one to obtain the penalty value.

**Justification for SEAMO**

SEAMO was chosen for this application for a variety of reasons. First, SEAMO's stated objectives – finding a set of plans that are not only good, but as diverse as possible

– are the same as the objectives of the school redistricting problem. Since the final decision making in this process is done by school officials, finding diverse plans is almost as important as finding good plans, since diverse plans help officials see the tradeoffs between different solutions.

SEAMO is able to find good plans with a relatively small population size compared to other algorithms, without resorting to fitness sharing or global dominance calculations. This makes it suitable for school redistricting because each solution takes up a considerable amount of memory and evaluating plans more than necessary can be computationally expensive. The school district has almost 300 neighborhoods, and each evaluation function must examine them all at least once.

SEAMO has been shown to outperform many other popular EMO algorithms as the size of the search space increases. This, too, makes it suitable for school redistricting, since the search space of this problem, for $s$ schools and $p$ polygons, is on the order of

$$s^{(p-s)}$$

(since polygons that contain a school must be assigned to that school). Since there are at least 12 schools at every level and almost 300 polygons, the search space is very large even if unreasonable solutions (such as those that assign non-contiguous regions to schools) are ignored.

Finally, SEAMO was chosen because it is easily extensible. Since it does not involve assigning weights to evaluation functions and only needs to know which solution dominates another, it will be easy to change or add new evaluation functions in the future. This is important, because several new evaluation functions, such as region contiguity and feeder statistics, are in development, and more are likely to be added in the future. Due to SEAMO's simplicity, it should be possible to add these functions without altering the algorithm in any way.

**Adaptation of SEAMO to School Redistricting**

The SEAMO for School Redistricting (SEAMO-SR) algorithm takes as input an initial plan to use as a *seed plan* for the first generation. A plan can be thought of as a group of *regions*, each of which contains a set of polygons assigned to one school. SEAMO-SR begins by generating an initial population with the seed plan. To do this, it creates $g$ copies of the seed plan, where $g$ is the population size of one generation. For each copy, it selects one polygon on the border between two regions. It then grows a *sub-region* by starting with the selected polygon and adding new polygons adjacent to the current sub-region, with a probability $n$ of stopping each time a new polygon is added. Thus, the expected size of a sub-region is $1 + 1 / n$. All polygons in the sub-region are then swapped to one school chosen at random.

Once the initial population is generated, SEAMO-SR performs selection, crossover, mutation and replacement on the population for 500 generations, or until ten generations pass with no replacement. SEAMO-SR's implementation of these evolutionary processes follows.

**Selection:** As in SEAMO, each member of the population is automatically selected to produce a child with one other, randomly selected member. Thus, in each generation, each member is expected to be selected for crossover twice.

**Crossover:** The child solution in SEAMO-SR is, initially, a copy of its first parent solution. Then, for each school in the plan, there is a probability of 0.5 that the second parent's region for that school will be copied into the child. Thus, the child plan will typically be similar to both parents, a quality that is essential for convergence in an EA [4].

**Mutation:** Each child is given a single mutation. In SEAMO-SR, mutation involves growing a sub-region around a border polygon and swapping it to a school chosen at random, in the same way that members of the initial population are generated. However, sub-regions in mutation stop growing with probability $m$. Since the members of the initial generation are meant to vary more from the seed plan than children are meant to vary from their parents, it is typically the case that $n < m$.

**Replacement**: Replacement in SEAMO-SR is conducted in a manner nearly identical to SEAMO. If a child is found to dominate one of its parents, it replaces that parent; if the child dominates both of its parents, it randomly chooses which one to replace; if the child does not dominate either parent, it is discarded. The only difference between replacement in SEAMO is that children with the best recorded value on any objective are not automatically preserved. This was omitted because plans that maximize any one criterion tend to be very poor, overall.

When SEAMO-SR finishes its evolutionary search, it selects all of the non-dominated members of the final population and returns them in random order.

**Search Methods Previously Implemented**

desJardins et al. implemented and compared several methods for heuristic local search in their previous work [4]. These methods were basic hillclimbing, biased hillcliming with blind bias, and biased hillclimbing with diversity bias.

**Basic Hillclimbing:** Given a seed plan, this method randomly chooses a polygon and swaps it to a random school. If this change results in a lower weighted sum of the evaluation criteria, it is kept, otherwise it is discarded. When no change that results in a lower weighted sum of evaluation criteria can be made to a plan, it is said to be a *local minimum* and is returned by the search. If more plans than one plan is desired, the process starts over again from the seed plan.

**Biased Hillclimbing with Blind Bias:** Given a seed plan, this method randomly chooses a polygon and swaps it to a random school. If this change results in a plan that dominates the original plan, the change is kept; if the original plan dominates the new plan, the new plan is discarded. However, if neither the new plan or the original plan dominates the other, the new plan is added to a list of *incomparable* plans. When it is not possible to randomly swap a polygon and produce a dominant plan, the plan reached is returned as a local minimum. If more plans are desired, the process begins again with a plan randomly selected from the incomparable list.

**Biased Hillclimbing with Diversity Bias:** This method is the same as biased hillclimbing with blind bias, but the plan selected from the incomparable list is the one with the greatest Euclidean distance in evaluation space from the returned plan.

| | Seed | n=.1, m=.2 | n=.05, m=.2 | n=.05, m=.1 | n=.2, m=.3 | n= .3, n=.2 |
|---|---|---|---|---|---|---|
| F1 | 0.709 | 0.535 (0.026) | **0.529** (0.013) | 0.563 (0.022) | 0.538 (0.016) | 0.548 (0.027) |
| F2 | 0.257 | **0.225** (0.002) | 0.232 (0.002) | 0.233 (0.003) | 0.227 (0.003) | 0.227 (0.003) |
| F3 | 0.236 | **0.213** (0.003) | 0.215 (0.002) | 0.218 (0.003) | 0.215 (0.003) | 0.215 (0.002) |
| F4 | 0.125 | 0.094 (0.009) | 0.097 (0.010) | 0.121 (0.021) | 0.095 (0.017) | **0.093** (0.008) |
| F5 | 0.104 | 0.070 (0.010) | 0.070 (0.009) | 0.098 (0.015) | **0.068** (0.010) | 0.072 (0.011) |
| Total | 1.433 | **1.139** (0.033) | 1.145 (0.025) | 1.236 (0.052) | 1.145 (0.036) | 1.156 (0.030) |
| Diversity | | 0.053 (0.002) | 0.051 (0.004) | **0.078** (0.013) | 0.068 (0.022) | 0.061 (0.006) |

***Table 1*** *– Average values of $f_1 \ldots f_5$, totals, and diversity measurements for seed plan and different values of n and m. Standard deviations are shown in parentheses.*

## Empirical Results

I conducted three experiments. The first was designed to find the optimal values of *n* and *m* for SEAMO-SR; the second was designed to determine how long different values of *n* and *m* take to converge; and the third was a comparison of SEAMO-SR to the heuristic local search methods previously implemented by desJardins et al. [4].

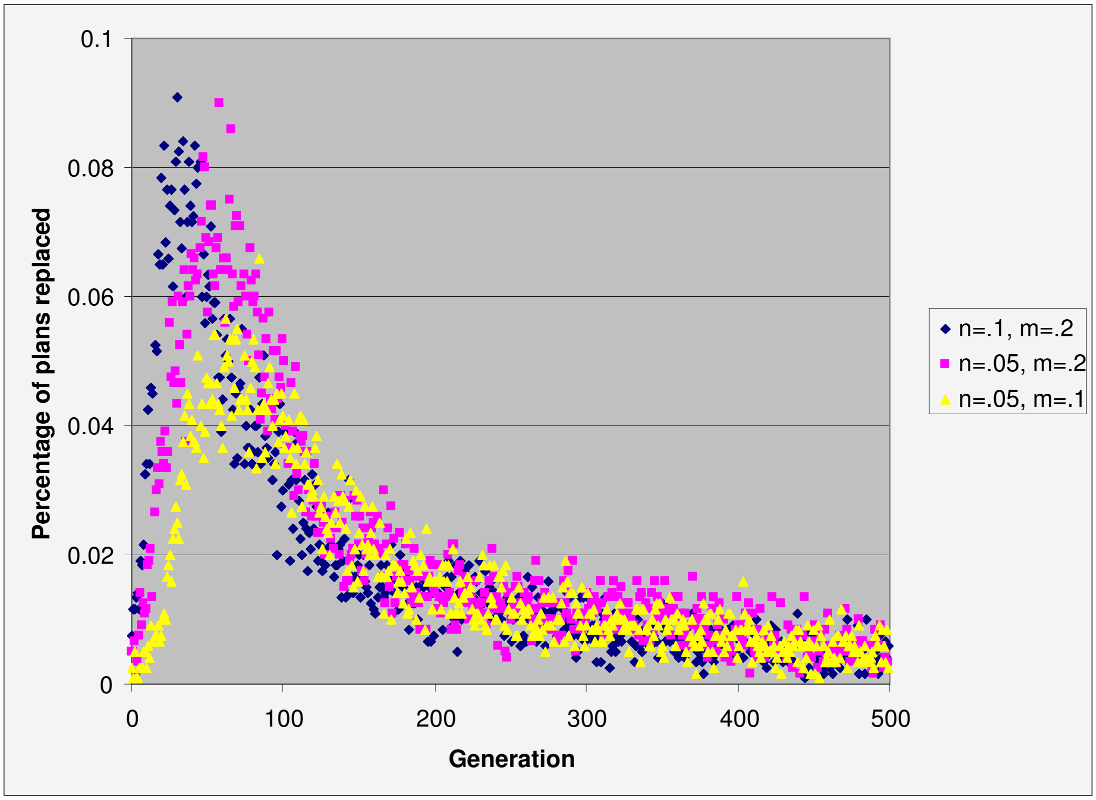
### Varying *n* and *m*

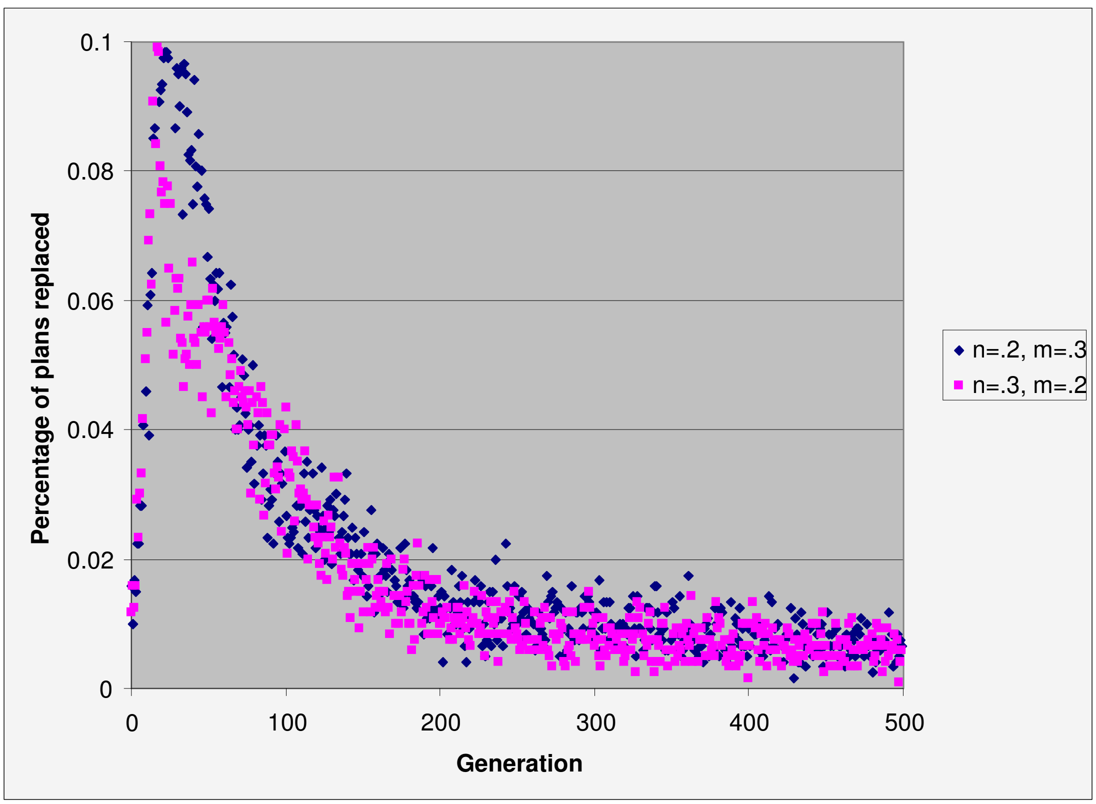
Recall that *n* is the probability that a sub-region will stop growing when the initial population is being generated, and *m* is the probability that a sub-region will stop growing during mutation. Since there are few guidelines on how to set rates of mutation in EAs, optimal values for *n* and *m* had to be reached empirically.

The results of this experiment may be found in Table 1. The seed plan used was the current middle school plan for Howard County, Maryland. This plan was chosen because it uses the most recent data obtained for the school district, and because the middle school level is the next level that will go through the redistricting process in Howard County. This data was obtained by generating groups of 30 plans using SEAMO-SR with different values of *n* and *m*, which are shown on the table.

The values in the “F1” through “F5” rows in Table 1 reflect the average value of the corresponding evaluation function across the 30 plans generated for each pair of values of *n* and *m* (with the exception of the “Seed Plan” column, which reflects the evaluation function values of the original plan). The “Total” row represents the total of the previous five rows. Recall that lower values are better for these rows. The “Diversity” row contains the average Euclidean distance between plans for each group of 30 plans. For this row, higher values are better.

These results indicate that varying *n* and *m* can have significant results on the quality of plans found. Setting these values too low seems to produce particularly poor results, as the group of plans produced with *n*=.05 and *m*=.1 had by far the worst results. However, even this group of plans still produced an average evaluation total significantly lower than the seed plan. The other groups of plans had similar evaluation totals, with the group produced by *n*=.1 and *m*=.2 producing the best results. Increasing or decreasing *n* or *m* from these values seems to cause a steady decrease in plan quality. Diversity was similar across all groups of plans, with the *n*=.05, *m*=.1 group having the highest diversity. However, this is probably due to the fact that members of this group were, in general, further from the Pareto front, and thus further from one another.




***Figures 1 and 2 –*** *Average percentage of population replaced at each generation for varying values of n and m.*

| | Best SEAMO-SR | Biased Hillclimbing with Blind Bias | Biased Hillclimbing with Diversity Bias | Basic Hillclimbing | Seed Plan |
|---|---|---|---|---|---|
| F1 | 0.535 (0.026) | 0.471 (0.009) | 0.637 (0.211) | **0.375** (0.011) | 0.709 |
| F2 | 0.225 (0.002) | **0.209** (0.001) | 0.197 (0.020) | 0.248 (0.004) | 0.257 |
| F3 | 0.213 (0.003) | 0.196 (0.001) | **0.180** (0.018) | 0.223 (0.003) | 0.236 |
| F4 | **0.094** (0.009) | 0.137 (0.001) | 0.290 (0.056) | 0.147 (0.008) | 0.125 |
| F5 | **0.070** (0.010) | 0.130 (0.006) | 0.306 (0.063) | 0.126 (0.012) | 0.104 |
| Total | 1.139 (0.033) | 1.144 (0.011) | 1.611 (0.197) | **1.121** (0.020) | 1.433 |
| Diversity | 0.053 (0.002) | 0.024 (0.001) | **0.298** (0.043) | 0.058 (0.001) | |

***Table 2 –*** *Average values of $f_1 \ldots f_5$, totals, and diversity measurements for seed plan, SEAMO-SR, and previously implemented search methods. Standard deviations are shown in parentheses.*

### Convergence for varying *n* and *m*

While the groups of plans in the previous experiment were being generated, I tracked the percentage of the population being replaced in each generation. Since SEAMO relies primarily on replacement to improve the quality of its population, this should provide another estimate of how well each method performs. The results of this experiment may be found in Figure 1 and Figure 2 (they have been broken up for easier readability).

Figures 1 and 2 show the average amount of replacement that occurred in each generation for the different groups of plans. More total replacement (i.e., a greater area under the "curve" of each group) should correspond to better total plan quality. These results reflect the results of the previous experiment; most groups except $n$=.05, $m$=.1 perform similar amounts of total replacement, and these groups have similar plan qualities. The $n$=.05, $m$=.1 group appears to perform considerably less replacement, however, and its plan quality is indeed lower.

### Comparison of SEAMO to previous search methods

In order to compare SEAMO-SR to the previous search methods developed by desJardins et al. [1] (basic hillclimbing, biased hillclimbing with blind bias, and biased hillclimbing with diversity bias), I generated 30 school plans with each of these methods using the original middle school plan as a seed.

The results of this experiment are presented in Table 2. The best values obtained by SEAMO-SR (i.e., the column from Table 1 with the lowest total $f$-values) are duplicated and compared with the average values of the plans produced the three hillclimbing methods. Their performance mirrors the performance shown in previous work: basic hillclimbing and biased hillclimbing with blind bias produce the best results but with low diversity, while biased hillclimbing with diversity bias produces highly diverse plans at the expense of plan quality [1]. SEAMO-SR's performance, at first glance, appears comparable to the first two hillclimbing methods. While its plan quality and diversity is between biased hillclimbing with blind bias and basic hillclimbing, all three values are extremely close compared to those achieved by biased hillclimbing with diversity bias.

Note, however, that SEAMO-SR is the only method that has better $f$-values than the seed plan across *all* evaluation functions. Since $f_1$, $f_2$, and $f_3$ have relatively high values compared to $f_4$ and $f_5$, the hillclimbing methods tend to improve these criteria at the expense of the others. This tends to lead to plans with better school utilizations and

FARM and MSA distributions, but high busing costs and students who could walk to a school being bused to one further away. While SEAMO-SR does not make such significant gains on the first three evaluation functions, it is also able to improve busing costs and walk area usage. Thus, the average plan generated by SEAMO-SR dominates the seed plan, which gives it a significant advantage over the hillclimbing methods.

## Conclusions and Future Work

The Simple Evolutionary Algorithm for Multiobjective Optimization (SEAMO) developed by Mumford-Valenzuela [2] has been successfully applied to the domain of school redistricting. SEAMO is a good fit for this application because it finds plans that are both high-quality and diverse, because it does well in applications with large search spaces, and because it endeavors to conserve memory and computation time. The implementation of SEAMO was refined via empirical results, and the final version of the algorithm was compared to previous heuristic local search methods developed by desJardins et al. [1] for this application. SEAMO was found to produce results comparable to the previous methods in terms of plan quality and diversity. However, SEAMO was shown to improve all evaluation criteria simultaneously, while the previous methods tended to improve some at the expense of others. This gave SEAMO a significant advantage, since the plans that it generates are more likely to dominate the plan used to seed the search.

In the future, I plan to alter the SEAMO adaptation to preserve plans with the best individual function values, and compare this approach to the one presented above. However, most future work for this application will consist of adding or changing evaluation functions. Functions which evaluate region compactness (i.e., how much each region resembles a circle) and feeder statistics (i.e., what portion of students who attend a lower-level school stay together when they move to a higher-level school) are currently in development, and several others are being considered. Additionally, some of the evaluation functions are being re-normalized to have values more comparable to the others (for instance, $f_1$ is being generally reduced, and $f_{4,5}$ are being generally increased). It should be interesting to see how this impacts SEAMO's performance with respect to the hillclimbing methods.

## References

[1] Abraham, A. and Jain, L, " Evolutionary Multiobjective Optimization," Evolutionary Multiobjective Optimization, Ed. Abraham, A., Jain, L., and Goldberg, R., Springer, 2005, pp. 1-6.

[2] Coello Coello, C, " Recent Trends in Evolutionary Multiobjective Optimization," Evolutionary Multiobjective Optimization, Ed. Abraham, A., Jain, L., and Goldberg, R., Springer, 2005, pp. 7 - 31.

[3] Marie desJardins, Blazej Bulka, Ryan Carr, Andrew Hunt, Priyang Rathod, and Penny Rheingans , *Heuristic search and information visualization methods for school redistricting*. Proceedings of the Eighteenth Conference on Innovative Applications of Artificial Intelligence (IAAI-06).

[4] Mumford-Valenzuela, C, " A Simple Approach to Evolutionary Multiobjective Optimization," Evolutionary Multiobjective Optimization, Ed. Abraham, A., Jain, L., and Goldberg, R., Springer, 2005, pp. 55 - 79.