DECIMATION OF RIVER GEOMETRY DATASETS USING GENETIC ALGORITHMS FOR USE IN SURFACE-WATER MODELS

Charles Berenbrock, Hydrologist, U.S. Geological Survey, Reston, VA, ceberenb@usgs.gov

Abstract

Surface-water hydraulic models require accurate representation of the river and (or) floodplain geometry, and the resulting dataset can be too large for most one-dimensional models. Selecting the appropriate data points to use in the model from among the hundreds or thousands of data points can be both challenging and tedious.

The problem is even more challenging for multi-dimensional datasets such as bathymetry or datasets produced from using Light Detection and Ranging (LiDAR). These datasets are typically used in two- and three-dimensional surface-water models. The standard procedure usually consists of gridding, which generalizes the terrain—missing the high and low elevations. To more effectively perform this task, the Genetic Algorithm (GA) computer program was modified to decimate (i.e. reduce) multi-dimensional datasets. The program was then used to decimate data for a hypothetical example and data taken from an actual bathymetric and LiDAR dataset. Results indicated that the program successfully reduced the data. Terrains produced by the GA are fairly representative of the original data, and volumetric differences from the original terrain were smaller for the GA produced terrain than standard procedures of decimating LiDAR. Results also showed that near-optimal results could be obtained in a single GA run.

INTRODUCTION

Traditionally, surveyed stream profiles and cross sections and (or) Light Detection and Ranging (LiDAR) scans are used to obtain data that describe the channel shape of streams and floodplains and are used in mathematical computer models to simulate flow hydraulics and sediment transport in a stream. A cross section is a series of data pairs (distance and elevation) along a straight line that is roughly perpendicular to streamflow. These datasets can be large. For example, data for approximately 500 cross sections were collected on the Kootenai River in northern Idaho. The number of data points for each cross section ranged from about 500 to more than 2,000 points (Barton et al., 2004; and Moran and Berenbrock, 2003). LiDAR and bathymetric datasets present an even larger problem. These datasets are usually used in two- and three-dimensional surface-water models. For example, a raw LiDAR dataset from the Lower Coeur d'Alene River for a 1 kilometer (km) by 1 km area consisted of more than 350,000 data points (x, y, and z). If a 10 km x 2 km reach of this river and floodplain were selected to be modeled, the dataset would consist of more than 6 million data points.

For flood insurance studies, the Federal Emergency Management Agency (FEMA) indicates that cross-section points should be located at breaks in the ground slope and should approximate the actual shape of the channel and (or) floodplain (FEMA, 1995). There is no point minimum as long as the actual shape of the channel and floodplain are well defined. The FEMA requirement applies to cross-section data, but is a reasonable requirement for multi-dimensional datasets such as digital elevation models (DEM) and bathymetric and LiDAR datasets.

These large datasets can be reduced to smaller, less dense datasets that are easier to work with—a process called decimation. Previous investigators have developed automatic decimation procedures. Chen and Guevara (1987) presented an automatic point selection procedure called "very important points" (VIP) for selecting points directly from DEMs or triangulated irregular networks (TIN). VIP is essentially a high-pass filter that selects data based on the distance a point is from the 4 lines connecting its diametrically opposed neighbors. Factors such as slope and proximity guide the selection of data points. This procedure has several potential problems. First, the peak of a small, sharp hill will be considered more important than a peak of one that is large, yet slopes gently. Secondly, the VIP procedure chooses nearly all of the points along valleys and ridges. It is desirable to capture these important features. However, if a ridge or valley follows a straight line, that feature may be represented by two or a few points instead of by many points. Another disadvantage is that the VIP procedure chooses nearly all of the points along the boundary, thereby overly defining the margins of the study area.

Berenbrock (2006) developed a genetic algorithm (GA) for decimating cross-section data while ensuring the integrity of the cross-section geometry. The program successfully decimated cross-section data and fit better to the original data than standard procedures—selection of a data point a set distance apart or selecting every 10th, 20th, or n^{th} point. On average, differences between the original cross sections and the GA-produced cross sections were about 30 percent less than cross sections obtained using standard procedures.

The purpose of this paper is to describe application of a GA to the decimation of LiDAR data and demonstrate that the GA is a viable approach. The GA described in this paper uses a single objective optimization scheme for decimating LiDAR and bathymetric data. A hypothetical example and a case study with actual data are presented to validate the genetic algorithm. The hypothetical example is a square dataset composed of 961 x, y, and z data points. The case study, Coeur d'Alene River and Floodplain Application, consists of bathymetric data from the river and LiDAR data from the floodplain.

GENETIC ALGORITHM

Genetic algorithms apply the ideas of Darwin's theory of evolution: individuals more adapted to the environment have a better chance to survive ("survival of the fittest"). Genetic operators such as selection, reproduction (crossover), and mutation are used to improve a population. Holland (1975) was the first to apply these operators. Since then, GAs have been applied successfully to many water resource problems (McKinney and Lin, 1992; Cieniawski et al., 1995; Vink and Schot, 2002; Chen, 2003; Knaapen and Hulscher, 2003). Vink and Schot (2002) and Chen (2003) indicated that GAs are capable of handling highly nonlinear, discontinuous, nondifferentiable, interdependent, and nonconvex problems where many other techniques such as linear and nonlinear programming, heuristic, etc. cannot. Simulated annealing relies on a weighted objective function that only finds one optimal solution per iteration, whereas, GAs are able to find multiple convex or nonconvex solutions in a single iteration (Cieniawski et al., 1995).

In a GA, a population of individuals is created and evolved until an individual is obtained that best represents the salient features of the dataset or until a specified number of generations is met. Individuals exhibit traits that can be inherited. For the problem at hand, traits are combinations of bathymetry and LiDAR data (terrain) points that are either included or excluded (removed) from the dataset to create individuals. The inclusion or exclusion of traits might be represented as a string of numbers or chromosomes. For example, the binary string "10101111" might represent an individual where the second and fourth chromosomes (data points) are excluded (a value of 0) and the other chromosomes included (a value of 1). The fitness of an individual (thus a fitness of traits or data point combinations) is represented as a value from the fitness or objective function.

Chromosomes are passed from parents to offspring through a process called "crossover" in which randomly selected chromosomes of the parents are combined or swamped to create children. For example, two individuals (parents) with traits 10101010 and 11111111 are crossed at the sixth through the eighth chromosome to produce two children, 10101111 and 11111010.

Individuals with superior fitness values are more likely to be allowed to reproduced, although this rule is often relaxed or altered to increase the genetic diversity of the population. Additional diversity is also introduced through mutation in which randomly selected chromosomes of the children are changed (reassigned) to create a combination of traits not present in either parent. A child that has been mutated may exhibit better or poorer fitness than the parents. Mutation rates are set very low.

Repeating the selection, crossover, and mutation processes over many generations under conditions controlled by a fitness function results in better-fit individuals and the overall population improves. The concept is described as pseudo code instructions in figure 1. Instructions within the "do-loop" are repeated until some time (number of generations) has elapsed, a threshold criterion has been met, or best individual fitness has reached a plateau. Goldberg (1989), Grefenstette (1990), Davis (1991), and Mitchell (2002) provide more complete discussions of the GA concept.

- Generate a random population, a collection of strings
- Do for some time
  - Evaluate each individual (string) in the population using the fitness function [f(i)]
  - Select two individuals (parents) from the population according to their fitness (the better the fitness, the greater the chance to be selected)
  - Allow for crossover–the parents form two new individuals (children). If no crossover was performed, the children are exact copies of the parents
  - Allow for mutation
  - Place children in new population
- End do

Figure 1. Pseudo code for a simple genetic algorithm.

PROGRAM DESCRIPTION

The cross section GA program (Berenbrock, 2006) was generalized and modified to create a new binary GA program capable of decimating LiDAR and (or) bathymetric data. The program uses strings of ones (1s) and zeros (0s) to represent the traits of individuals (combinations of present or omitted data points) from the dataset. The position of chromosomes within the strings and the lengths of the strings are fixed on the number of original data points (n). Ones (1s) are inserted into chromosome positions corresponding to data points that define the convex hull for the dataset. A convex hull is a set of points that define the extent or boundary of the dataset in n-space.

An individual represents a collection of data points with values of x, y, and z on the terrain. The GA program uses TINs to represent or describe the terrain because data points can be irregularly spaced, whereas in a DEM, points must be regularly spaced. A TIN is composed of three points to form a triangular arrangement. An individual's volume is the volume beneath the terrain or volume of the TINs, which is calculated by computing the volume in each TIN and summing the TIN volumes.

The program computes an individual's fitness as the difference between the individual's volume and the original volume. A limit on the number of included data points was accomplished by imposing a two-conditional fitness function (Berenbrock, 2006). The two-conditional fitness function is:

$$f(i) = \begin{cases} \left| \left(\sum_{j=1}^{n} v_j \right) - v_{original} \right| & \text{if } inc_i \leq plimit \\ \left| \left(\sum_{j=1}^{n} v_j \right) - v_{original} \right| \cdot 10^{(inc_i - plimit)} & \text{otherwise} \end{cases} \tag{1}$$

where f is the value of the fitness for individual i, n is the number of TINs in individual i, v_j is the volume of the TIN for trait j, $v_{original}$ is the volume of the original dataset, inc_i is the number of included data points in individual i, and plimit (point limit) is the maximum number of points to be included in the decimated dataset. The first condition applies if the number of included points in an individual is less than or equal to the point limit; otherwise, the second condition applies. The GA then minimizes the fitness function (*Minimize* f(i)) to identify the best-fit individual or optimal dataset.

To reduce the number of generations and program run time, an initialization technique was used to generate the initial population. This technique sets the probability of a data point being included to the plimit divided by the string length (n). By narrowing the search space, the GA then focuses its search on approximately the right number of points and does not explore unlikely solutions (including all points, including no points, and so on) (Berenbrock, 2006).

The GA is generational; two elite individuals are copied each generation. Tournament selection is used as the reproduction methods, and tournament size is 3. In the validation test, several different crossover rates (Pc), population sizes, and number of generations at first were used. The mutation rate (Pm) was set to 1/n, a standard rate suggested by Reed *et al.* (2000, 2003).

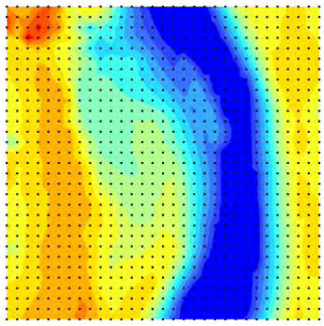
PROGRAM VALIDATION

Data representing a hypothetical example and data taken from actual bathymetric and LiDAR datasets collected on the Coeur d'Alene River and Floodplain were used to validate the binary GA. The plimit for the hypothetical example was arbitrarily set to 15 percent of the hypothetical data (144 data points) and 10 percent for the Coeur d'Alene data. A mutation rate of 1/n was used (Reed *et al.*, 2000, 2003).

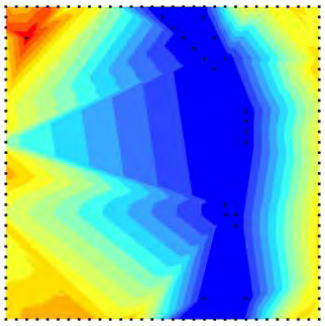
Hypothetical Example

The hypothetical dataset consists of 961 regularly spaced points (x, y, z) on a 31 x 31 grid, spaced 16 m apart (fig. 2A). The GA was run 10 times with a population size of 80 individuals, for 100 generations, using a crossover rate of 0.30 and plimit of 15 percent (144 data points). The number of included data points ranged from 139 to 144 (table 1). Because this GA seeks to minimize the fitness function (Eqn. 1), the run with the lowest fitness value is the superior or "best" run for the given fitness function and domain. In this case, Run 8 had the superior fitness value (166 m3) given the 10 runs. Even though the GA relies on randomness in sampling and in creating the initial population, the range in best fitness for all runs was small as compared to the range in average fitness and root mean square error (RMSE) fitness. The small range in best fitness suggests that near optimal results could be obtained in a single GA run.

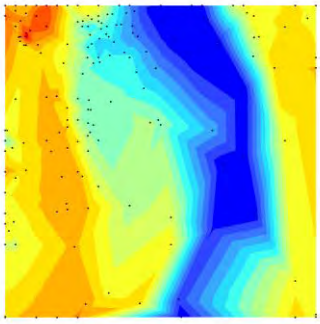
Figure 2 shows a colorized relief terrain representation of the TIN produced by the hypothetical (original), VIP, LATTICETIN and 10 GA runs. VIP and LATTICETIN are two commonly used TIN procedures. The number of points used in VIP and LATTICETIN was not allowed to exceed the plimit (144 points). From a qualitative viewpoint, the VIP run (fig. 2B) is a very poor representation of the original dataset under this condition; only 20 points were used to define the interior terrain while 120 points were used to define the boundary. The terrain is under emphasized in the interior and over emphasized at the boundary. As discussed earlier, VIP retains most if not all of the boundary points in its solution. The LATTICETIN run (fig. 2C) preserved the major features fairly well but also had inaccurate features near the boundaries especially near the stream. For example, the stream near the southern boundary extended too wide probably because only a few data points are used to define the terrain in this area. For the LATTICETIN run, 37 data points are located on the boundary. In the 10 GA runs, the major topographic features are preserved fairly well (fig. 2D through 2M). Inaccurate features are also seen near the boundaries especially near the stream. This is probably due to the GA runs containing fewer points on the boundary (ranging from 19 to 28 points) which is less than the LATTICETIN run. The fewer points especially near the stream caused the stream to not fully extend to the boundary of the study area or caused the stream to extend too widely, similar to the LATTICETIN run. For example, the stream in runs 1 (fig. 2D), 2 (fig. 2E), and 10 (fig. 2M) did



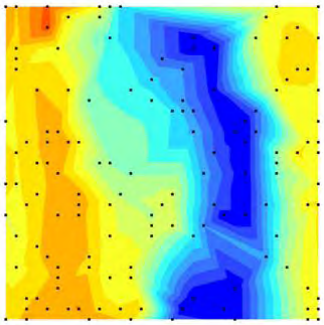
A. Original, n=961 (120)



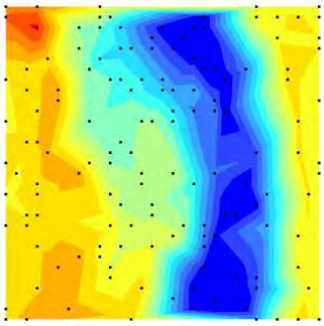
B. VIP Run, n=144 (120)



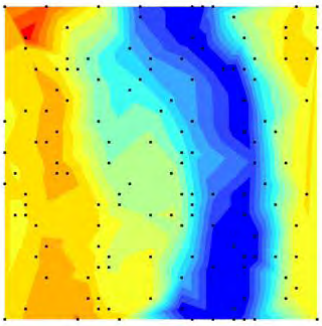
C. LATTICETIN Run, n=144 (37)



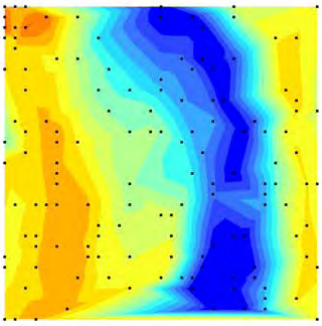
D. GA Run 1, n=140 (28)



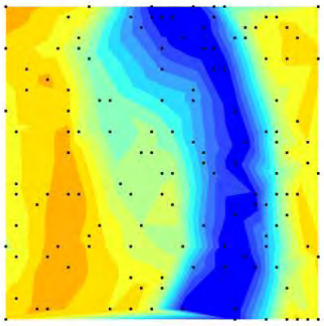
E. GA Run 2, n=144 (25)



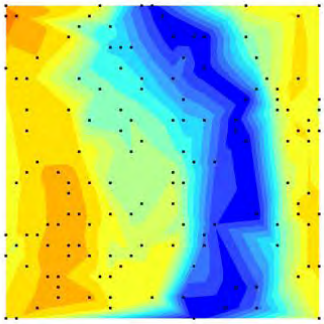
F. GA Run 3, n=141 (23)



G. GA Run 4, n=140 (21)



H. GA Run 5, n=139 (20)



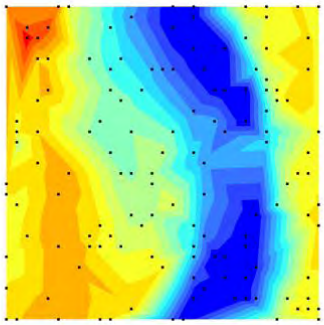
I. GA Run 6, n=141 (19)

Explanation

Elevation of TIN, in meters

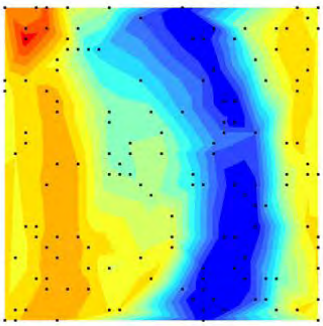
0 - 0.5	2.5 - 3	5 - 5.5
0.5 - 1	3 - 3.5	5.5 - 6
1 - 1.5	3.5 - 4	6 - 6.5
1.5 - 2	4 - 4.5	6.5 - 7
2 - 2.5	4.5 - 5	7 - 7.5

• Data point (x, y, z)

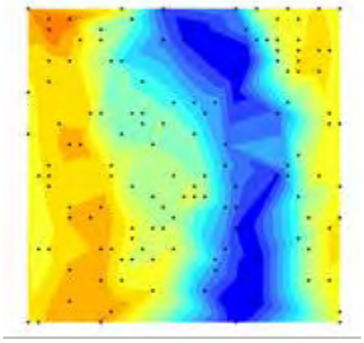


J. GA Run 7, n=140 (26)

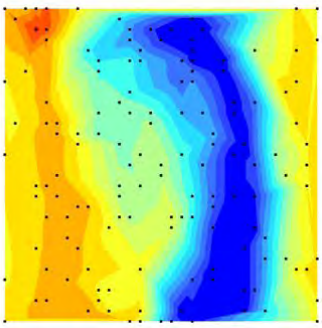
Figure 2. Terrain from the original dataset, VIP run, LATTICETIN run, and GA runs for the hypothetical example. (n is the number of data points, and value inside the parenthesis is the number of points located on the boundary)



K. GA Run 8, n=139 (21)



L. GA Run 9, n=143 (19)



M. GA Run 10, n=140 (20)

Figure 2. — Continued.

Table 1. Best fitness value, number of data points, and number of points on the boundary for the hypothetical example.

Run	Number of data points	Best fitness value (cubic meters)	Number of points on boundary	Figure no.
Original	961	--	120	3A
VIP Run	144	--	120	3B
LATTICETIN Run	144	--	37	3C
[1]GA RUN 1	140	361	28	3D
[1]GA RUN 2	144	355	25	3E
[1]GA RUN 3	141	771	23	3F
[1]GA RUN 4	140	518	21	3G
[1]GA RUN 5	139	271	20	3H
[1]GA RUN 6	141	260	19	3I
[1]GA RUN 7	140	332	26	3J
[1]GA RUN 8	139	166	21	3K
[1]GA RUN 9	143	1,110	19	3L
[1]GA RUN 10	140	460	20	3M

[1] The crossover rate (P_c) was set to 30 percent, the mutation rate (P_m) was set to 1/n, plimit was set to 15 percent or 144 points, and n is the number of data points in the original data (961).

not fully extend to the northern boundary of the study area, and the stream in run 4 (fig. 2G) did not fully extend to the southern boundary.

The best, average, and RMSE fitness values for Run 8 are shown in figure 3. The best fitness decreased as the number of generations increased indicating that the program functioned correctly for this large dataset; best fitness decreased from 1,540 m^3 to 518 m^3. To measure the distribution of the population, the average and RMSE fitness values are calculated for each generation. If the average and RMSE fitness values are large, then the diversity between individuals is high; if average and RMSE fitness values are small, the diversity is low. The genetic algorithm might not perform well if the diversity is too high or too low. For example, if

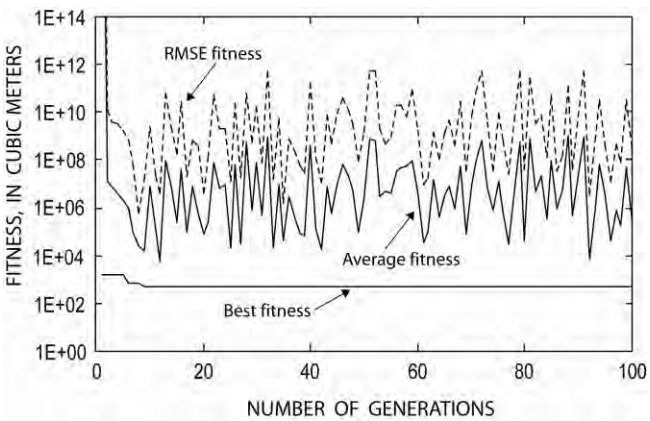


Figure 3. Best fitness, average fitness, and root mean squared error (RMSE) fitness for each generation of GA Run 8 for the hypothetical example.

the population's diversity is too low, the genetic operator "crossover" becomes almost ineffective and the population probably will have a hard time escaping from the local optimum where the population has converged too. On the other hand, if the population's diversity is too high, an optimal solution might not be reach or take a long time (number of generations) to reach. The large range in average and RMSE fitness values shown in figure 3 indicates that the population is highly diverse. These values also fluctuated from one generation to the next indicating changes in the population. Similar results were observed in the other nine runs.

Volumetric results from the VIP, LATTICETIN, and GA runs and the original are shown in figure 4. The volume for the original and each run is calculated by computing the volume for each TIN in the dataset and then summing those TIN volumes. Results for the VIP run were much lower than the original indicating that the VIP-produced terrain is inaccurate in this example. The volumetric differences from the original for this run are quite large (fig. 4B). The number of interior data points (20 points) in this run was not enough to produce a reasonable volumetric representation of the original. Volumetric results for the LATTICETIN run closely track the original (fig. 4A). Differences in volume from the original and the LATTICETIN run are small for heights greater than 1 m (fig. 4B); differences increase when heights are less than 1 m, which can be seen in the unnatural shape of the stream (fig. 2C). Results for the GA runs showed that volumes closely track the original and are bunched together in a narrow band (fig. 4A), which also supports that near optimal results could be obtained in a single GA run.

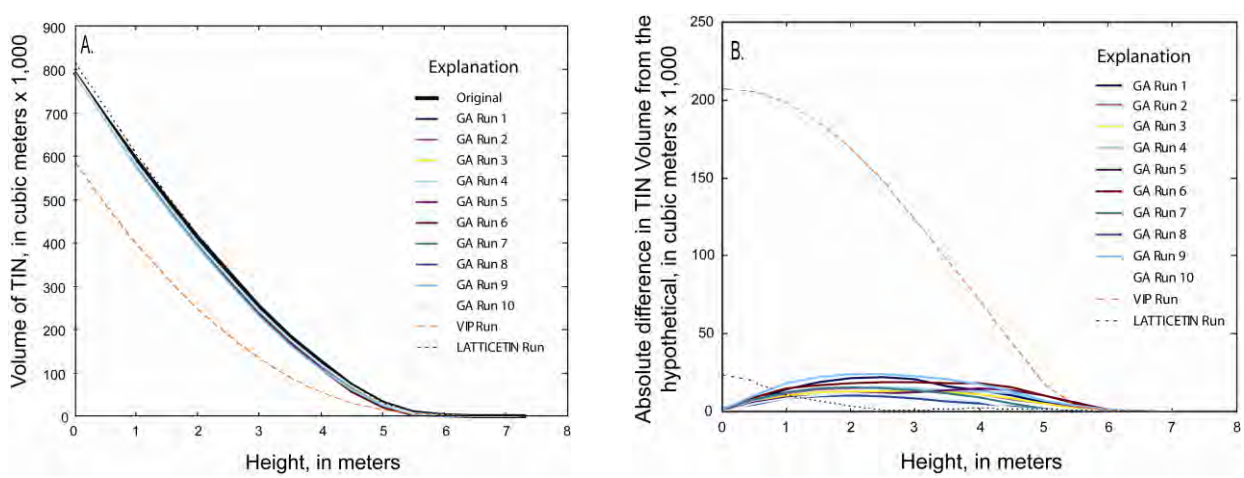


Figure 4. (A) TIN volumes for the original, VIP, LATTICETIN, and GA runs and (B) volumetric differences from the original for the hypothetical example.

Differences in volume from the original in the GA runs are also small (fig. 4B). The greatest differences occurred in the middle heights and are probably cause by the fitness function in the GA basing its volumetric calculations only at a 0 m height. These differences are small but are evidence of the deficiency in the GA. A fitness function is needed that integrates the entire volume along the curve.

Coeur d'Alene River Application

For the real world application, a section was extracted from LiDAR and bathymetric datasets from the Coeur d'Alene River and Floodplain near river mile 156 (Berenbrock and Tranmer, 2008). The subsection, 1073 m (0.67 mi) wide and 952 m (0.60 mi) long, contains 10,080 points. The GA initially was run with the same parameter values as in the hypothetical example, except plimit was arbitrarily set to 10 percent (1,008 points).

The GA was run 10 times for the Coeur d'Alene River dataset. The best fitness and number of points for each run are shown in table 2. The run with the lowest fitness value is the superior or "best" run because this GA seeks to minimize the fitness function (Eqn. 1). The range of best fitness for the GA runs varied from 165 m^3 to 1,390 m^3. The fourth GA run had the superior fitness of all the runs. The best, average, and RMSE fitness values for Run 4 are shown in figure 5. Again, as expected, the best fitness decreased as the number of generations increased indicating that the program functions correctly for large datasets. The best fitness decreased from 12,200 m^3 to 165 m^3. The large range in average and RMSE fitness values indicates that the population is highly diverse, and the fluctuations from one generation to the next show the changing diversity in the population. Similar results were observed in the other nine runs.

Table 2. Best fitness value, number of data points, and number of points on the boundary for the Coeur d'Alene River application.

Run	Number of data points	Best fitness value (cubic meters)	Number of points on boundary	Figure no.
Original	10,080	--	396	6A
VIP Run	1,010	--	262	6B
[1]GA RUN 1	1,003	473	52	--
[1]GA RUN 2	993	1,310	48	--
[1]GA RUN 3	990	1,390	41	--
[1]GA RUN 4	1,003	165	41	6C
[1]GA RUN 5	1,008	903	41	--
[1]GA RUN 6	1,008	856	45	--
[1]GA RUN 7	962	1,360	50	--
[1]GA RUN 8	1,005	242	46	--
[1]GA RUN 9	1,008	812	47	--
[1]GA RUN10	1,004	940	45	--

[1] The crossover rate (P_c) was set to 30 percent, the mutation rate (P_m) was set to 1/n, plimit was set to 10 percent or 1,008 points, and n is the number of data points in the original data (10,080).

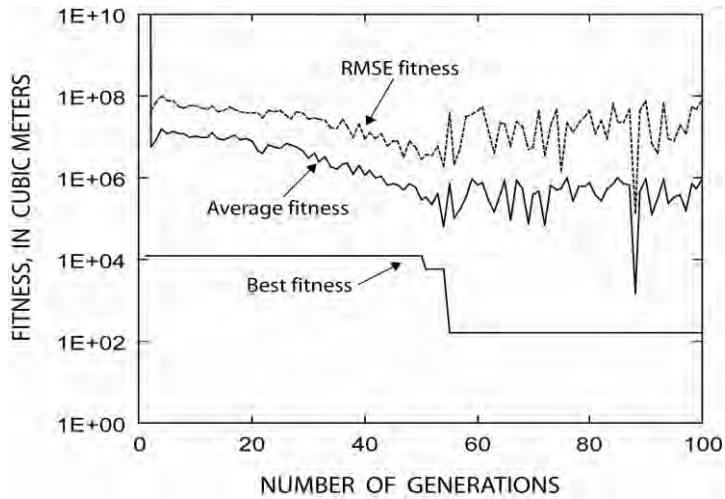
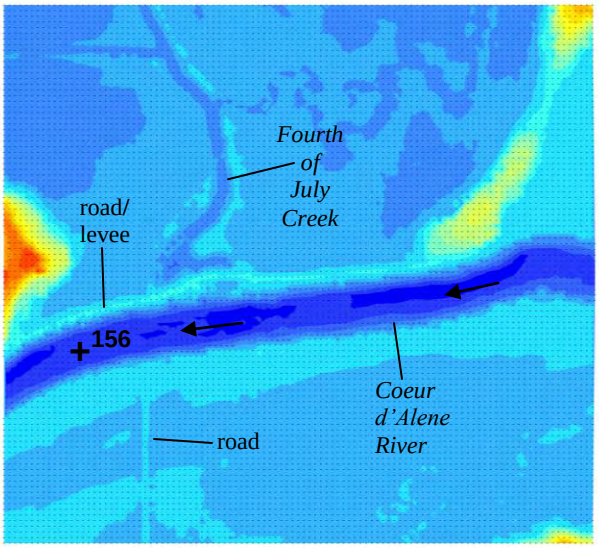


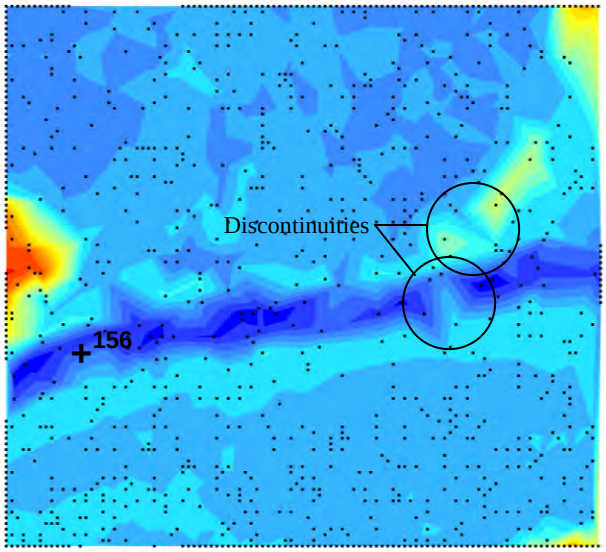
Figure 5. Best fitness, average fitness, and root mean squared error (RMSE) fitness for each generation of GA Run 4 for the Coeur d'Alene River.

Figure 6 shows a colorized relief terrain representation of the TINs produced by the original dataset, VIP run and the GA run 4. From a qualitative viewpoint, the major topographic features for the VIP run are generally preserved (fig. 6B). However, several large discontinuities in the terrain occurred in the study area. The discontinuity in the river could have significant impacts on river flows if this dataset were used in a multi-dimensional model, and the discontinuity of the hill could have impacts on floodplain flows. Neither the Fourth of July Creek, the road/levee north of the river, nor the road south of the river is distinguishable (fig. 6B). If the VIP had fewer points located along its boundary (262), there would be more points available for the interior that would give more definition to the interior especially to areas in the river. A LATTICETIN run was not conducted because the LATTICETIN requires regularly spaced data throughout the domain. The LiDAR and bathymetric datasets used in the subsection are regularly spaced but the datasets do not line up to one another to create a regularly spaced dataset. Together the dataset is considered irregularly spaced, and thus, a LATTICETIN run could not be performed. This is a distinct advantage of a TIN and a disadvantage in LATTICETIN that requires regularly spaced data. The major topographic features for the fourth GA run are generally preserved even along the boundary (41 points located on the boundary) (fig. 6C) unlike what happen in the hypothetical example. A plausible explanation is that there are enough points near the boundary to obtain a good representation. Neither the Fourth of July Creek nor the road/levee north of the river is distinguishable in the fourth GA run, but the road south of the river is somewhat distinguishable (fig. 6C).

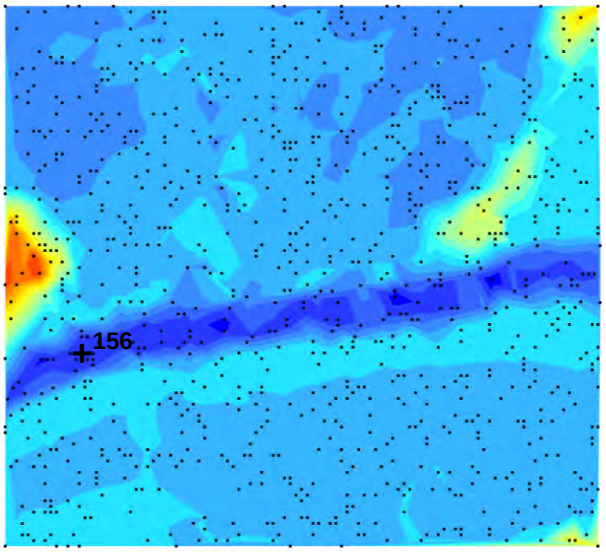
Volume was calculated for these runs. These volumetric results also closely tracked the original. Again, results from all GA runs fell within a narrow band similar to results from the hypothetical example, which supports that near-optimal results could be obtained in a single run. Volumetric differences were also calculated for these runs, but only the VIP run and the fourth GA run, the superior GA run, are shown in figure 7. Differences from the original in the VIP run were greater at all heights than in the fourth GA run. The largest differences between the two runs occurred when the height ranged from 10 m to 20 m.



A. Original, n=10,080 (396)



B. VIP Run, n=1010 (262)



C. GA Run 4, n=1003 (41)

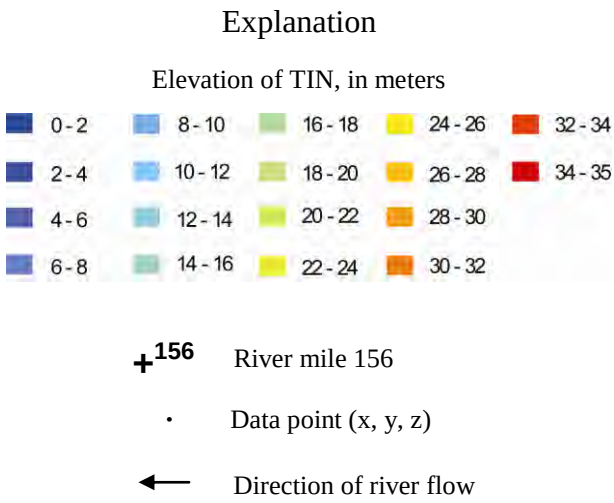


Figure 6. Terrain from the original, VIP run and GA run 4 for the Coeur d'Alene River application. (n is the number of LiDAR points, and number inside parenthesis is the number of points located on the boundary)

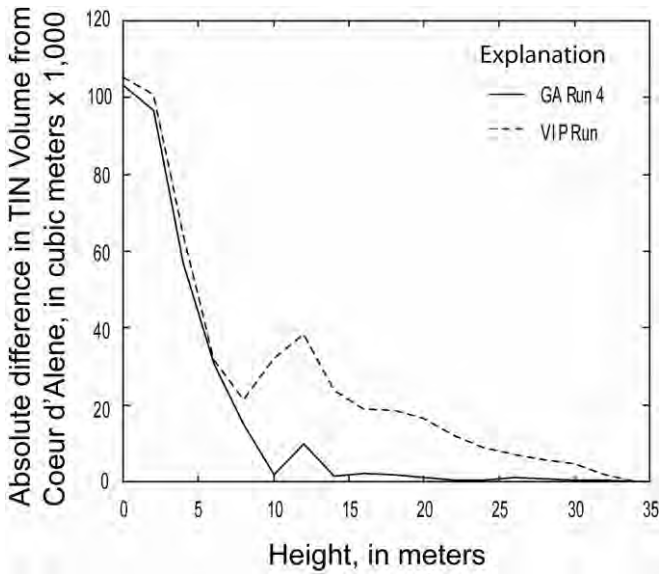


Figure 7. Volumetric differences in TIN volumes from the original to the VIP Run and GA Run 4 for the Coeur d'Alene River application.

SUMMARY AND DISCUSSION

This paper demonstrates that a Genetic Algorithm (GA) is a viable approach for solving the LiDAR/bathymetric decimation problem. GAs cannot assure exact solutions, but yield reasonable solutions to optimization and search problems. For the hypothetical example, TINs from the GA runs are fairly representative of the original, but gave poor representation near the boundary. The VIP compared poorly to the original and to the GAs. The LATTICETIN compared favorably to the GA runs. The GA and LATTICETIN runs compared favorably to the original data with the LATTICETIN, overall, closer to the original. To better fit the original data, the fitness function used in the GA needs to integrate the entire volumetric curve and rather than just the total volume of the TINs. Also results from these GA runs demonstrated that near-optimal results could be obtained in a single GA run.

For the case study of Coeur d'Alene River and Floodplain, TINs from the GA runs are fairly representative of the subsection even along the boundary. The VIP run also showed fair representation. Some of the finer features such as creeks, levees and roads were poorly defined in the GA and VIP runs. TIN volumes from the superior GA run (no. 4) and VIP compared favorably, with the GA run having a smaller difference from the original. A LATTICETIN run could not be performed because the combined bathymetric and LiDAR data do not line up to produce a regularly spaced dataset.

Although the genetic algorithm was successful in decimating the datasets, it still needs to be tested with datasets having more data points. The current fitness function calculates the volume of an individual at the zero height. A fitness function that integrates volume along its height (hypsometric curve) might cause the GA to select better fit individuals in the population that have smaller differences from the original at all elevations. Also the current fitness function always selects the plimit as the number of inclusions. A fitness function that gradually penalizes

more points and gradually gives credit to fewer points might also enable the GA to select a good minimum number of points. These improvements are left for future investigations.

REFERENCES

Barton, G.J., Moran, E.H., Berenbrock, C. (2004). Stream Channel Cross Sections for the Kootenai River Between Libby Dam, Montana, and Kootenay Lake, British Columbia, Canada. U.S. Geological Survey Open-File Report 2004-1045, p 35.

Berenbrock, C. (2006). "A genetic algorithm to reduce stream channel cross section data," Journal of the American Water Resources Association, 42(2), pp 387-394.

Berenbrock, C., and Tranmer, A.W. (2008). Simulation of flow, sediment transport, and sediment mobility of the Lower Coeur d'Alene River, Idaho. U.S. Geological Survey Scientific Investigations Report 2008-5093, p 164.

Chen, L. (2003). "Real Coded Genetic Algorithm Optimization of Long Term Reservoir Operation," Journal of the American Water Resources Association (JAWRA) 39(5), pp 1157-1165.

Chen, Z.T., and Guevara, J.A. (1987). "Systematic Selectin of Very Important Points (VIP) from Digital Terrain Model for Constructing Triangular Irregular Networks," Proceedings of the Eighth International Symposium on Computer-Assisted Cartography, N.R. Chrisman (Editor), pp 57-67.

Cieniawski, W.E., Eheart, J.W., and Ranjithan, S. (1995). "Using Genetic Algorithms to Solve a Multiobjective Groundwater Monitoring Problem," Water Resources Research 21(2), pp 399-409.

Davis, L., (1991). "Hybridization and Numerical Representation," In: Handbook of Genetic Algorithms, L. Davis (Editor). Van Nostrand Reinhold, United Kingdom, pp 62-72.

FEMA (Federal Emergency Management Agency). (1995). Guidelines and Specifications for Study Contractors. Federal Emergency Management Agency, Publication 37, U.S. Government Printing Office, Washington, D.C., p 174.

Goldberg, D.E., (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Pub. Co., p 372.

Grefenstette, J.J. (1990). "Genetic Algorithms and Their Application," In: Encyclopedia of Computer Science and Technology, A. Kent and J.G. Williams (Editors). Marcel Dekker, New York, New York, 21(6), pp 139-152.

Holland, J.H. (1975). Adaptation in Natural and Artifical Systems. University of Michigan Press, Ann Arbor, Michigan.

Knaapen, M.A.F., and Hulscher, S.J.M.H. (2003). "Use of a Genetic Algorithm to Improve Predictions of Alternate Bar Dynamics," Journal of Water Resources Research 39(9), 1231, doi:10.1029/2002WR001793, 2003.

McKinney, D.C., and Lin, M.D. (1992). "Design Methodology for Efficient Aquifer Remediation Using Pump and Treat Systems," In: Mathematical Modeling in Water Resources, T. Russel et al. (Editors). Elsevier Science, New York, New York, pp 695-702.

Mitchell, M. (2002). An Introduction to Genetic Algorithms. The MIT Press, Cambridge, Massachusetts, (8th printing), p 209.

Moran, E.H. and Berenbrock, C. (2003). "GPS—Time Saver and Functional," U.S. Geological Survey Western Water Watch, 1(1), pp 6-7.

Reed, P., Minsker, B. and Goldberg, D.E. (2000). "Designing a Competent Simple Genetic Algorithm for Search and Optimization," Journal of Water Resources Research, 36(12), pp 3757-3761.

Reed, P.B., Minsker, B., and Goldberg, D.E. (2003). "Simplifying Multiobjective Optimization – An Automated Design methodology for the Nondominated Sorted Genetic Algorithm-II," Journal of Water Resources Research, 39(7), pp 1196-1206.

Vink, K. and Schot, P. (2002). "Multiple-Objective Optimization of Drinking Water Production Strategies Using a Genetic Algorithm," Journal of Water Resources Research 38(9), pp 1157-1165.