Abstract – The problems in river hydraulics is too complicated to develop exact, generalized empirical or regression equations. Regression analysis has been used extensively in the past to address these problems but owing to its inadequacies pertaining to a first-hand functional form determination and clustering effect of influential points and groups of points that could even have been erroneous, regression has failed to deliver satisfactory results. With the advent of huge computational resources, it has been possible to develop statistical models based purely on analysis of historical data. Alongside, the development of the new paradigm of evolutionary algorithms, based on the modus operandi of a plethora of natural phenomena, has enabled the modern modeler to develop very precise models which are able to explain the variability in data to a suitably satisfactory level and then predict the state variables corresponding to the combination of circumstances that are entirely new to the model developed. Genetic programming is one such paradigm which is free of many limitations of regression, empirical and dimensional analyses and gives the flexibility of the choice of parameters of the algorithm in a robust manner that allow the model for best description of variability as well as reasonable predictability. Present work uses genetic programming for flow prediction in vegetated channel. The developed model shows very high predictability of flow resistance.

Keywords – flow resistance, genetic programming, river hydraulics, vegetation.

I. INTRODUCTION

Channel vegetation refers to vegetation that typically are emergent aquatic plants or herbs, trees and shrubs that exist inside and in proximity to water. They may be submerged completely or non-submerged (emergent), have different densities, height of submergence, stem-flexibility, stem geometry, surface characteristics and spacing. These characteristics influence flow and morphology of a channel in different ways. Submerged and emergent vegetation cause different velocity profiles in a channel, governed by their height and flexibility.

Literature is replete with numerous works on flow-vegetation interactions. Huang and Nanson [1] found that vegetation affected channel hydraulic geometry exponents. Other, specific case studies such as those of Malkinson [2], McKenney [3] and Erskine [4] have also found the effects of riparian vegetation on channel properties like cross-sectional characteristics of streams, channel morphogenesis and channel widening. Channel mobility decreases with increase in vegetation density. It has been reported [5, 6] that deviation of the velocity profile in the outer layer over a gravel bed with vegetation cover on the walls is much larger than the case of flow over a gravel bed without vegetation cover on the walls. Studies have also shown that vegetation spacing affects turbulence by changing the three dimensional flow patterns [7]. Experiments in laboratory flumes have been carried out to quantify the flow-vegetation interactions by various researchers [8, 9]. Green [10] has done the field measurement in natural vegetated fields. Also, detailed numerical simulations of flow through vegetation were performed [11, 12]. Several empirical [13] and theoretical relations [14, 15] have been proposed to describe the flow-vegetation interactions. Bennett [16] designed a flume-based study to alter the flow pattern within a straight, degraded stream corridor by using simulated emergent vegetation of varying density placed at key locations within the channel. Their study showed that flow velocity can be markedly reduced within and near the vegetation zones, flow can be diverted toward the opposite bank, and vegetation density controlled the magnitude of these effects.

Huthoff et al. [17] have ascertained that because of simplicity, empirical equations have better field applicability than theoretical ones. Galema [18] based on data available in the literature has compared different predictors of flow characteristics. Based on the comparison of the predicted and measured values for both conditions, it can be concluded that no simple predictor exists for both conditions [18]. Empirical equations have a general problem that they are applicable for restricted circumstances, those in which the equations have been formulated. In most of the cases, an empirical study is carried out to grasp the trend of the dependence of resistance on the measurable vegetation parameters and an equation is developed that explains the observed behavior most accurately possible in the given context. The equation developed is then
used to predict behavior of the system, given the values of independent parameters. This methodology lacks in the generalization of behavior of the physical phenomena being explained. Other than empirical techniques, Computational Fluid Dynamics (CFD) models have been used to solve the Navier-Stokes equations in specified domains. Due to computational limitations, vegetation characteristics, most importantly the vegetation geometry has to be kept simple enough, leading to idealizations that differ much from actual vegetation characteristics.

A lack of proper understanding of such natural phenomena in have led hydrologists to employ techniques of soft computing, for tasks like non-linear data modeling, which make it possible to predict parameters based on the significant recognizable variables. Over the last two decades, soft computing methods such as artificial neural networks (ANN), genetic algorithm (GA), fuzzy logic and particle swarm optimization (PSO) have encountered a growing interest in hydrological studies and have been applied as powerful alternative computational tools. ANNs were used to predict scour holes around bridge piers by Lee [19]. They were also used for rainfall forecasting by French [20] and by Lee [21] to carry out long term tidal predictions. GA and fuzzy control were applied to a combined sewer pumping station by Yagi [22] and a groundwater management problem was addressed using genetic algorithm by Sidiropoulos [23]. Genetic algorithm was used for predicting hourly-based flow discharge hydrographs from level data by Tayfur [24]. Penalty-type genetic algorithm was used to guide rational reservoir flood operation by Chang [25]. ANN models for monthly streamflow time series prediction were developed by Wu [26]. Other recent developments came in the paradigm of soft computing technology with the introduction of variety of optimization algorithms, including the PSO, by Kennedy & Eberhart [27]. Since then PSO has been used extensively as an alternative optimization algorithm. A PSO based simulation–optimization model for the solution of groundwater management problems was proposed by Gaur [28]. PSO was also applied for the optimization of gravity dam and sluice gate by Wu et al. [29].

The paradigm of genetic programming is a descendent of genetic algorithms but differs fundamentally from it in approach and structure. While genetic algorithms are used to search for solution values directly, genetic programming aims to find programs that yield best solutions. It is also evident from the search space of genetic programming that it designs and looks for expressions and not values, although numeric constants are calculated by multi-gene regression procedure after the genetic programming searches for the best population.

Genetic programming is a relatively new domain in soft computing and has gained popularity in a variety of applications, including those in river hydraulics and sediment dynamics in fluvial systems. Azamathulla [30] used linear genetic programming for discharge prediction in compound channels. Kisi et al. [31] developed suspended sediment models using genetic programming. Aytek [32] attempted sediment modeling using a genetic programming approach.

The primary objective of the present work is to develop the flow predictor for vegetative channel using the multi-gene genetic programming approach using historical datasets from flume experiments with vegetation. This is done by first selecting optimum parameters for the algorithm and then assessing model efficiency with the help of several criteria.

II. GENETIC PROGRAMMING

The basic search strategy behind genetic programming [33] is a genetic algorithm, which imitates biological evolution. It differs from this traditional genetic algorithm in that it typically operates on parse trees instead of bit strings. GP initializes a population consisting of the random members known as chromosomes (individual) and the fitness of each chromosome is evaluated with respect to a target value. The principle of Darwinian natural selection is used to select and reproduce “fitter” programs. GP creates equal or unequal length computer programs that consist of variables (terminal) and several mathematical operators (function) sets as the solution. The function set of the system can be composed of arithmetic operations (+,−, /, *) and function calls (such as _ex , x, sin, cos, tan, log, sqrt, ln, power). Each function implicitly includes an assignment to a variable, which facilitates the use of multiple program outputs in GP, whereas in tree-based GP those side effects need to be incorporated explicitly [34]. The fitness of a GP individual may be compared by using the equation

$$f = \sum_{j=1}^{N} \left( |X_j - Y_j| \right)$$  \hspace{1cm} (1)

Where Xj is the value returned by a chromosome for the fitness case j and Yj is the expected value for the fitness case j.

An initial population of expressions is generated with randomized selection procedure. A fitness function is used to assess the individual expressions. It is usually the deviation of the model output from the actual output, or, the error, which must be minimized. An algorithm based on Darwinian model of reproduction and survival of the fittest and genetic recombination is used to create a population of individuals from the current population. A part of the parent population is filtered based on their fitness values and are used to create new offspring population which replaces the old generation. Then, each individual is assessed for fitness and the process is repeated. This algorithm produces populations which, over generations, tend to exhibit average improving fitness and adapt themselves to the changes in their environment.
Individuals in the population

The structures that undergo adaptation are expressions containing functions, operators, (independent) variables and numeric coefficients, whose form, size and complexity can dynamically change during the process. The search space of expressions is the set of all possible composition of expressions that can be formed using recursively from the available set of ‘n’ functions F= \{f_1, f_2, f_3, ..., f_n\} and an available set of m terminals T= \{a_1, a_2, a_3, ..., a_m\}. While the terminals are either variables, numeric constants and universal constants (e.g. pi), the functions comprise of mathematical operators and standard mathematical functions. It should be noted that the set of functions and terminals used in a particular problem should be selected so as to be capable to solve the problem.

The search space

The search space consists of the hyperspace of all valid expressions that can be formed using the available set of functions and terminals. It is important to decide the maximum level/depth of the expressions since at times, the later generations may lead to complicated expressions which are long in length and take much more time to compute and may cause the algorithm to be slow. To allow the genetic operators like mutation and crossover to happen, the tree data structure used to represent and store each expression. The formation of an expression involves choosing a function first and then choosing another function or terminal as the parameters of the original function.

The fitness function

The available dataset of input and output combinations provides the algorithm to assess the fitness of an individual expression created. In effect, the expression created by the algorithm can be evaluated for the inputs specified and the output yielded can be compared with the actual correct output in the dataset. The same is carried out for each of the data points and the sum total of the errors serves as an indicator of fitness in an inversely related sense. The closer the sum of errors is to zero, the better is the expression assessed.

Operations that modify the expressions

The genetic programming algorithm operates on the population of expression using some operations. These utilize the structure of the expressions and are essentially to create, select and breed populations to further create new individuals. The individual operations are described in this section.

Fitness proportionate reproduction

This operation copies the best individuals from the parent population to the offspring population. The proportion of the population that has to be copied is specified as a parameter of the algorithm. This procedure ensures that if an individual has a sufficiently high fitness value, it can survive though generations.

Crossover or Recombination

This operation is responsible for creating variations in the population by producing expressions in the offspring generation that combine traits from two parent individuals. The process starts with choosing two parent individuals proportionate to their fitness values from the entire parent population and then combines parts of expressions from each individual at varying/randomized positions. The result is a combination of two crossover fragments, one from each parent individual.

The state of the system

At any stage, the state contains the individuals of one (current) generation only. The best result, if needed to be found, must come from the current population only, since there is no bookkeeping of individuals from the past.

Algorithm termination

The result of the algorithm is the best individual from the state of the system (the current populations). The algorithm may be terminated when a previously specified fitness has been reached, depending on the tolerance values, or when the specified number of generations have worked. The single best individual is considered to be the output of the algorithm.

Parameters in the algorithm

The genetic programming algorithm operates on populations of symbolic expressions by creating and changing them. The following parameters decide the extent of such operations.

Population size

It is the number of expressions in a generation. This value remains constant throughout the algorithm run.

Number of generations

This parameter indicates the number of iterations of the algorithm, where in each iteration new sets of individuals are created using the operators as mentioned above.

Maximum number of genes

In multi-gene genetic programming, more than one expression are formed using genetic programming algorithm followed by linear regression of the expressions in the final population is carried out to yield a composite weighted expression list that fits the data best. For this, since one gene represents one expression, the maximum number of genes implies the maximum number of expressions in the final expression.

Maximum depth of genes

The gene stores an individual expression. The depth of a gene indicates the level of compositions of functions utilized for framing expressions. A gene with higher depth represents more convoluted expressions than one with lower depth. A limit on the depth would be
necessary to avoid very complicated expressions that again can increase complexity of the final expression and make the physical description non intuitive.

**Mutation-Crossover-Copy fractions**

The fractions of the entire population that must undergo the above three operations is another important operation that controls convergence rate of the algorithm to the actual best solution. By changing these fractions, one can control the rates in which variation takes place from one generation to the other. These must be decided and are utilized in every iteration to create the population of the offspring generation. The three values add up to 1.

**Tournament size**

The genetic programming algorithm utilized selection technique called tournament selection, where a set of population are chosen for creating the new generation based on a fitness criteria. The tournament size parameter is used to specify what percentage of the population shall be used to create new population.

**Elite fraction**

Elite fraction is that fraction of the initial (parent) population that will be copied without making any changes in them. This is done for the best individuals in the population and makes book keeping of older ‘good’ individuals unnecessary, since they will be copied every time a new generation is created.

**Build method**

The initial population creation can be done in three separate manners- ‘full’, ‘grow’ and ‘ramped-half-and-half’ methods. While in the ‘grow’ build method, if the maximum depth has not yet been reached, either a function or a terminal may be selected, causing more diverse tree structures with some branches longer than others, in the ‘full’ method, the trees created are of the full size and have been randomly created. The third method uses the first two in equal proportions.

**Parameter Selection**

The parameters of the algorithm have to be selected so that the best individual may be found. Unluckily, there are no general guidelines on parameter estimation yet and hence, a hit and trial procedure was used to reach to the best combination of parameters. Since the number of parameters is huge (a total of ten parameters in all), a stepwise selection of the parameters was carried out, where, a nominal value of other nine parameters being kept constant, the parameter to be optimized was scanned thoroughly across values and the results were compared to assess the effect of this one parameter only. For most of these parameters, the individual runs were repeated three times each to remove random correlations, and since the results varied each time.

### III. RESULTS AND DISCUSSIONS

Soft-computing needs functional form of a physical system describing its dependent and independent variables. Yen [35] has analyzed several equations in terms of their dependent and independent variables. The physical modeling of channel flow through vegetation is very complex. According to Lopez and Garcia [36], it is a function of many variables; the fluid properties, flow properties, vegetation characteristics and channel characteristics as follows:

$$f(u, h, k, i, m, C_d, D)=0 \quad (2)$$

where $u$ is the mean velocity, $h$ is the flow depth, $k$ is the height of the vegetation, $i$ is the channel slope. $D$ is the diameter of cylindrical vegetation and $m$ is the number of cylinders per $m^2$ horizontal area. $C_d$ is the non-dimensional drag coefficient. Symbolic regression was used to develop a flow predictor model as an expression relating the flow explicitly to the variables. Entire multi-gene genetic programming for modeling was implemented in MATLAB® software using "gptips", a genetic programming MATLAB® toolbox. Galema [18] documented a comprehensive database of different types of vegetations with hydraulic properties from different sources. Total number of observations used in the modeling is 445. Out of this, 345 values were used to train the model (model formation) and the remaining 100 values were used to test model performance. The parameters were varied in a range of values and the optimum values were recorded. Table 1 summarizes the results.

**Table 1: Optimum parameter values chosen**

| Parameter              | Optimum value               |
|------------------------|-----------------------------|
| Initial build method   | ‘ramped-half-and-half’      |
| Maximum depth          | 5                           |
| Maximum genes          | 7                           |
| Mutation ratio         | 0.5                         |
| Crossover ratio        | 0.3                         |
| Directly copied ratio  | 0.2                         |
| Tournament size (%)    | 60                          |
| Elite fraction         | 0.15                        |
| Population             | 1000                        |
| Generations            | 200                         |

Figure 1 Shows the variation of the correlation coefficient with changing population.
It is evident that the performance of the model asymptotically reaches a constant value and after the point when population was 1000, any further increase in population results in a nominal increase in performance. Thus, a population of 1000 was adopted for modeling. The model performance has been shown in the Figure 2. As it can be seen from the figure, model predicts the phenomena well.

IV. CONCLUSION

Vegetation-flow interactions are central to many problems of practical interest to hydraulic engineers including flood risk studies, sediment transport studies and the analysis of the hydraulic performance of river restoration schemes. Existing predictors are limited in terms of their applicability. Though a data driven technique like symbolic regression does not elucidate or utilize the physics of the system, it is primarily useful for sufficiently good prediction capabilities. Based on a large database, present work identifies a symbolic regression model using genetic programming based on statistical datasets. The generalization capacity of the model is very good considering so many different parameters affecting the flow-vegetation interactions.

V. REFERENCES

[1] H. Q. Huang, G. C. Nanson, “Vegetation and channel variation; a case study of four small streams in southeastern Australia,” Geomorphology, 18, 3–4, 237-249, 1997.

[2] D. L. W. Malkinson, “Scaling the effects of riparian vegetation on cross-sectional characteristics of ephemeral mountain streams—a case study of Nahal Oren, Mt. Carmel, Israel,” Catena, 69, 2, 103-110, 2007.

[3] R. McKenney, R. B. Jacobson, R. C. Wertheimer, “Woody vegetation and channel morphogenesis in low-gradient, gravel-bed streams in the Ozarks Region, Missouri and Arkansas, in Hupp, C.R., Osterkamp, W.R., and Howard, A.D. (eds), Biogeomorphology -- terrestrial and freshwater aquatic systems,” Geomorphology, 13, 175-198, 1995.

[4] W. Erskine, A. Keene, R. Bush, M. Cheetham, A. Chalmers, “Influence of riparian vegetation on channel widening and subsequent contraction on a sand-bed stream since European settlement: Widden Brook, Australia.,” Geomorphology, 147–148, 102-114, 2012.

[5] C. L. Jang, Y. Shimizu, “Vegetation effects on the morphological behavior of alluvial channels,” J. Hydr. Res. IAHR, 45(6), 763–772, 2007.

[6] H. Afzalimehr, S. Dey, “Influence of bank vegetation and gravel bed on velocity and Reynolds stress distributions,” International Journal of Sediment Research, 24, 2 , 236-246, 2009.

[7] T. Stoesser, S. J. Kim, P. Diplas, “Turbulent Flow through Idealized Emergent Vegetation,” Journal of Hydraulic Engineering , 2010.

[8] C. A. M. E. Wilson, T. Stoesser, P. D. Bates, A. Batemann-Pinzen, “Open channel flow through different forms of submerged flexible vegetation,” J. Hydraulic Eng. 129, 847– 853, 2003.

[9] A. Armanini, M. Righetti, P. Grisenti, “Direct measurement of vegetation resistance in prototype scale,” J. Hydraulic Res. 43, 481– 487, 2005.

[10] J. C. Green, “Effect of macrophyte spatial variability on channel resistance,” Adv. Water Res. 29, 426– 438, 2006.
[11] V. S. Neary, “Numerical solution of fully-developed flow with vegetative resistance,” J. Eng Mech. 129, 558–563, 2003.

[12] S. U. Choi, H. Kang, “Reynolds stress modeling of vegetated open-channel flows,” J. Hydraulic Res. 42, 3–11, 2004.

[13] N. Kouwen, M. Fathi-Moghadam, “Friction factors for coniferous trees along rivers,” J. Hydraulic Eng. 126, 732–740, 2000.

[14] B. M. Stone, H. T. Shen, “Hydraulic resistance of flow in channels with cylindrical roughness,” J. Hydraulic Eng. 128, 500–506, 2002.

[15] D. Poggi, C. Krug, G. G. Katul, “Hydraulic resistance of submerged rigid vegetation derived from first-order closure models,” Water Resour. Res. 45, W10442, 2009.

[16] S. J. Bennett, “Using simulated emergent vegetation to alter stream flow direction within a straight experimental channel,” Geomorphology, 44, 1–2, 115-126, 2002.

[17] F. Huthoff, D. C. M. Augustijn, S. J. M. H. Hulscher, “Analytical solution of the depth-averaged flow velocity in case of submerged rigid cylindrical vegetation,” Water resources research, Vol. 43, W06413, 2007.

[18] A. Galemia, “Evaluation of vegetation resistance descriptors for flood management,” Master Thesis, University of Twente, 2009.

[19] T. L. Lee, D. S. Jeng, G. H. Zhang, J. H., Hong, “Neural network modeling for estimation of scour depth around bridge piers,” Journal of Hydrodynamics, Ser. B, Volume 19, Issue 3, 378-386, 2007.

[20] M. N. French, W. F. Krajewski, R. Cuykendall, “Rainfall forecasting in space and time using a neural network,” Journal of Hydrology, 137, 1–4, 1-31, 1992.

[21] T. Lee, “Back-propagation neural network for long-term tidal predictions,” Ocean Engineering, 31, 2, 225-238, 2004.

[22] S. Yagi, “Application of genetic algorithms and fuzzy control to a combined sewer pumping station,” Water Science and Technology, 39, 9, 217-224, 1999.

[23] E. Sidirooulos, P. Tolikas, “Genetic algorithms and cellular automata in aquifer management,” Applied Mathematical Modelling, 32, 4, 617-640, 2008.

[24] G. Tayfur, T. Moramarco, “Predicting hourly-based flow discharge hydrographs from level data using genetic algorithms,” Journal of Hydrology, 352, 1–2, 77-93, 2008.

[25] L. Chang, “Guiding rational reservoir flood operation using penalty-type genetic algorithm,” Journal of Hydrology, 354, 1–4, 65-74, 2008.

[26] C. L. Wu, K. W. Chau, “Data-driven models for monthly streamflow time series prediction,” Engineering Applications of Artificial Intelligence, 23, 8, 1350-1367, 2010.

[27] J. Kennedy, R. Eberhart, “Particle swarm optimization,” Proceedings., IEEE International Conference on, 4, 1942-1948, 1995.

[28] S. Gaur, B. R. Chahar, D. Graillot, “Analytic elements method and particle swarm optimization based simulation–optimization model for groundwater management,” Journal of Hydrology, 402, 3–4, 217-227, 2011.

[29] X. Wu, Z. Qie, Z. Zhou, H. Zhang, “Application of improved PSO to optimization of gravity dam and sluice gate,” Intelligent Control and Automation, 2008.

[30] H. Azamathulla, A. Zahiri, “Flow discharge prediction in compound channels using linear genetic programming,” Journal of Hydrology, 454–455, 203-207, 2012

[31] Ö, Kisi, A. H. Dailir, M. Cimen, J. Shiri, “Suspended sediment modeling using genetic programming and soft computing techniques,” Journal of Hydrology,450–451, 48-58, 2012

[32] A. Aytek, Ö. Kişi, “A genetic programming approach to suspended sediment modeling,” Journal of Hydrology, 351, 3–4, 288-298, 2008

[33] J.R. Koza, “Genetic programming: On the programming of computers by means of natural selection,” A Bradford book, MIT Press, Cambridge, Mass, 1992.

[34] M. Brameier, W. Banzhaf, “A comparison of linear genetic programming and neural networks in medical data mining,” IEEE Trans. Evol. Comput., 5, 17–26, 2001.

[35] B. C. Yen, “Open channel flow resistance,” J. Hydraulic Eng., 128, 20–39, 2002.

[36] F. Lopez, M. Garcia, “pen-channel flow through simulated vegetation: Suspended sediment transport modeling,” Water Resour. Res. 34, 2341-2352, 1998.