Oil Sands Processability Analysis Using Symbolic Regression

Yixin Zhang[a,b,c,*], Qing Zhao[a,c], Zhenghe Xu[b,c]

[a] *Dept. of Electrical and Computer Engineering*
[b] *Dept. of Chemical and Material Engineering*
[c] *University of Alberta*

Abstract

Hot or warm water based bitumen production process from mineable oil sands is extremely complex in nature and highly sensitive to variability of oil sands ores and process conditions. Understanding ore processability and developing sensible markers for ore processability are considered to be a challenging task. In addition to processing variables such as temperature, hydrodynamics, process water chemistry and chemical additives, ore characteristics, such as bitumen content, connate water content and chemistry, fines content and more importantly types of fines play a decisive role in determining the processability of oil sands ores. It is therefore valuable to analyze the processability of oil sands ore using statistical modelling approaches. In this paper, a symbolic regression method based on genetic programming is applied to understanding oil sands ore processability, such as identifying sensible markers of ore processability. The analysis is conducted using variety input variables representing ore characteristics and operating conditions. The model is expressed analytically by a combination of these input variables and a given set of math operators and constants. This model provides a reliable prediction for the response variables, e.g, bitumen recovery. The results show an agreement between simulation and experiment data, highlighting the applicability of the Symbolic Regression (SR) method in identifying a mathematical model to describe the mechanisms involved in oil sands processing.

Keywords: Symbolic Regression, Genetic Algorithm, Genetic Programming, Kernel Methods, Oil Sands Processability, System Identification

1. Introduction

A significant portion of bitumen produced in Canada is from the mineable oil sands (6). Hot water based bitumen production process from mineable oil sands is extremely complex in nature and highly sensitive to variability of oil sands ores. Understanding ore properties and developing a sensible marker for ore processability have been proven to be highly desirable but challenging. Not only process variables but also ore characteristics play a decisive role in determining the processability of oil sands ores. There are three main contributing factors for poor performance of oil sands extraction(3): *(1)* lack of online determination of complex oil sands composition; *(2)* lack of advanced setup for process control; and *(3)* malfunctions or failures of mechanical equipment.

The current technology for improving oil sands processability is mostly based on single factor analysis or factorial design, which has not yielded satisfactory results. In recent years there have been considerable efforts and extensive development of machine learning and data driven techniques for process modeling and analysis. It is expected that these emerging techniques and knowledge will provide new means for tackling challenging tasks of oil sands process analysis. Using plant or Batch Extraction Unit (BEU) data, algorithms based on probabilistic programming and/or statistical data analysis have recently been applied to modeling oil sands ore processing(15).

Recently, machine learning methodology has initiated lots of academic and industry attempts in the field of engineering application. There are various research developments such as using evolutionary method finding re-entry vehicle path (10). In planning and manufacturing process, machine learning application are also widely used for building feature beasd design of computer aided process planning intelligent system (5). Researchers nowadays can even use evolutionary methodology to track a moving object using robot system (8). In machine learning field, evolution computing is a very specific subject and genetic algorithm (GA) can represent its essence.

Genetic Algorithm is a unique and useful family of techniques in data analysis. In recent years, there has been a significant amount of research on genetic algorithm that focuses on particular characteristics of data and whereat factors and their associations. Symbolic Regression (SR), for example, focuses on identification of a mathematical description on a hidden system from experimental data. The applications of SR algorithms have grown significantly during the past years. It has been shown that they could be a successful solution to dimensionality reduction modelling and optimization in a variety of areas including but not limited to environmetrics (4), microarray data analysis [5-7] document clustering (1), face recognition [9-11], blind audio source separation (13) and more. What makes SR algorithms particularly attractive is the function constraints

*I am corresponding author
Email addresses: yixin6@ualberta.ca (Yixin Zhang), qingz@ualberta.ca (Qing Zhao), zhenghe@ualberta.ca (Zhenghe Xu)

imposed on the factors they produce, allowing for better interpretability.

(remove and instead, explain how blockers are introduced and can self-evolve?)

In this work, the main objective is to identify significant markers for processability of mineable oil sands ores, based on which prediction results can be produced, which will potentially be integrated with the current industrial analysis system. In our approach, several sensible markers controlling the ore processability are identified, and analytical relationship between these markers and the oil recovery are to be established.

A preliminary SR framework for genetic algorithm suitable for prediction of oil sands ore processability analysis was proposed as shown in Figure 1. A greedy genetic algorithm is adopted to optimize the factors of the oil sands recovery prediction. The statistical mathematical characterization results are then obtained by SR. In order to extract features that are more meaningful, a modification to the SR algorithm is considered which incorporates certain partially known information/-constraints of the input attributes of given dataset. In this paper, SR algorithm is applied not only to simulated data (mainly for proof of concepts and verification), but also to the plant data for further validation and demonstration of the applicability.

2. Methodology

2.1. Symbolic Regression via Genetic Programming

In this paper, the symbolic regression (SR) via genetic programming (GP) is applied to analyzing oil sands processability. A genetic programming is an optimization procedure. From a given population X, it seeks item $x \in X$, which has the greatest fitness. A genetic programming searches for the best value by creating a small pool of random candidates, selecting the best candidates, allowing them to breed with minor variations, and finally repeating this process over many generations. These ideas are all inspired by the analogy to the evolution of living organisms(4).

A genetic programming typically includes:

- a genetic representation of candidates;
- a way to create an initial population of candidates;
- a function measuring the fitness of each candidate;
- a generation step in which some candidates die, some survive and others reproduce by breeding;
- a mechanism that recombines genes from breeding pairs and mutates others.

Symbolic regression (SR) is a method to search for a set of mathematical operators that identify an analytical description for the relationships among input and output attributes of the given data set. SR aims to extract an appropriate model from a space of all possible expressions $\mathcal{S}$ defined by a set of given

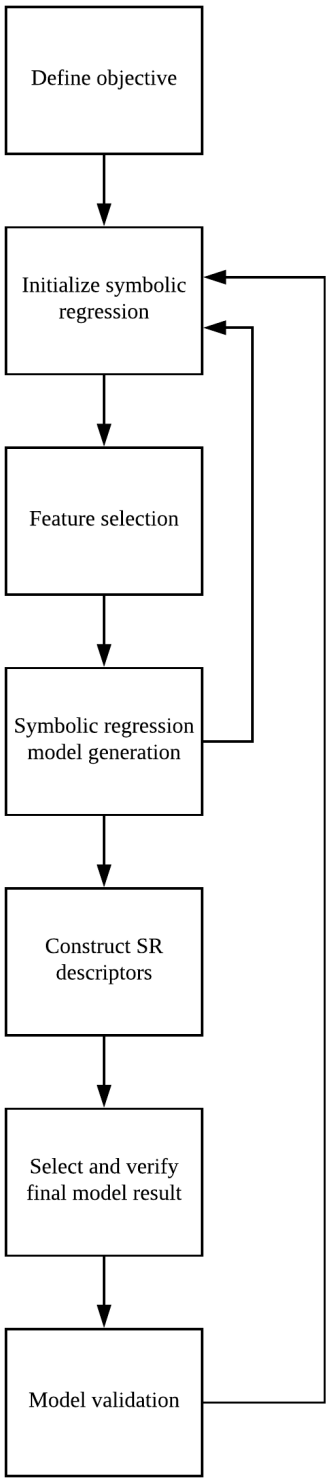

Figure 1: Genetic Programming(GP) iteration procedure: The tree-based GP iteration controls the evolved SR generation and destruct the unpromising offspring.

binary operations (e.g., +, −, ×, ÷, etc.) and mathematical functions (e.g., sin, cos, exp, ln, etc.), which can be described in the following nonlinear optimization problem:

$$f^* = \arg\min_{f \in \mathcal{S}} \sum_i \left\| f(\mathbf{x}^{(i)}) - y_i \right\|, \tag{1}$$

where $\mathbf{x}^{(i)} \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$ are input observations and output data, respectively. f is the unknown target model and f^* is the identified data model. SR is an NP-hard problem that can be solved using genetic algorithm. Moreover, a standard algorithm used for SR is GP, which is specialized for evolving generation and tree structures, e.g, searching for a space of mathematical expressions and minimizing various error metrics. Both the parameters and the form of equations are subject to search. In SR, many initial random symbolic equations compete via optimization to model experimental data in a most non-traditional manner. New equations are formed by reorganizing and combining previous equations and probabilistically varying their sub-expressions. The algorithm leads to equations that model the experimental data and at the same time the rejection of unpromising solutions. After an equation reaches a desired level of accuracy, the algorithm terminates, returning equations that may correspond to the intrinsic mechanisms of the original dataset(9). The overall procedure summarizing the above steps is shown in Figure 1.

In SR, the representative symbolic expressions are the combination of genes often represented as a binary tree of algebraic operations with numerical constants and symbolic variables as its leaves. Other more complex representations include acyclic graphs and grammars (11). The fitness of a particular equation is a numerical measure of how well it agrees with the data, such as the correlation of equation-generated results and the experimental data.

The expressions can vary among many mathematical operators, including *abs*, *exp* and *log* etc., or binary ones such as *add, sub, multiply*, and *divide*. If prior knowledge of the initial value is known, which is the so-called domain knowledge, the types of operations available can be chosen ahead of time. The terminal values consist of constants, and input attributes or function variables.

Mutation in a symbolic expression can change an operator in the binary tree, e.g, it can change the *add* functions into *sub*, change the arguments of an operation such as changing $x + c$ to $x + x$, delete an operation such as changing $x + x$ to x or add an operation by changing $x + x$ to $x + (x + x)$. If the operator is changed from a binary operation to a constant, one of the two sub-tree branches is discarded and those branches can be chosen randomly.

Crossover of a symbolic expression exchanges sub-trees in the binary trees of the initial expressions. For example, crossing $f(x) = x^2 + c$ and $f(x) = x^2 + sin(x) + x$ could produce a sub tree with $f(x) = x^2 + sin(x)$, from which the leaf node c was exchanged with the $sin(x)$ term.

2.2. SR Parameters and Constrained Optimization

In SR, once the fitness function is defined constrained optimization (with obtained symbolic expressions as the constraints) should be performed by treating every constant as the parameter. The constrained optimization can also be treated as a fitting problem by minimizing a cost function $Q(\alpha)$ (Equation 2), which can be chosen the sum of squared errors between the experimental data y and the prediction of the SR expression $f(x, \alpha)$. Such optimization can be evaluated by using algorithmic differentiation, also called automatic differentiation (AD) (7). AD is a useful tool in practical optimization problems involving calculation of first or high-order derivatives of the objective or constraint functions that are given in computer programs.

$$Q(\alpha) = \sum_{i=0}^{m} (y_i - f(x_i, \alpha))^2 \tag{2}$$

where x_i is input variable and α is model parameter.

One of the conditions for SR to work is that differentiable (smooth) functions such as logistic functions must be incorporated in the SR parsing tree, which encodes formula expression using combination of the identified markers. Otherwise optimization solutions with one or multiple sets cannot be achieved by the gradient calculation. The gradient of this SR model can generate a continuous search direction, whose information can be used for accelerating this entire process.

The constrained optimization is an iterative procedure which gradually improves the SR model quality using the gradient calculation, starting from initial parameters. As a start, the Jacobian matrices are evaluated, which are used to update the parameter vector for the iteration procedure to continue until a specified stopping criterion is reached. automatic differentiation is used for the calculation of all partial derivatives with respect to the parameter vector α. As a result, all constant values are extracted and replaced by an appropriate parameter α_i, given in Equation 3,

$$\nabla f = \left(\frac{\partial f}{\partial \alpha_1}, \frac{\partial f}{\partial \alpha_2} \ldots \frac{\partial f}{\partial \alpha_i} \right) \tag{3}$$

(the following paragraph together with Figure 2 need to be rewritten for better explanation)

The designed algorithm describing necessary steps for the proposed SR-based model is given as Algorithm 1. In the literature, the SR method has been utilized in many application areas. SR is able to understand the data and build a descriptive model. Most importantly it provides certain insights about the data and the model. Table 1 provides comparison among several data mining methods, including SR. In section 3, we demonstrate the application of SR to the analysis of oil sands ore properties.

3. Implementation Procedure

In this section, details on how to implement the SR-based algorithm are given first, and then as an example the algorithm is applied to a set of synthetic data which is generated to mimic the oil sands recovery process in a laboratory experiment.

Table 1. Comparison of SR with three other modeling techniques (Linear Regression, Neural Networks and Random Forests) which belong to three distinct culture in modeling prediction family (12).

.

	Linear Regression	Neural Networks	Random Forests	Symbolic Regression
Knowledge about explicit model structure required	Yes	No	No	No
Parametric or Non-parametric	Param.	Param.	Non-param.	Non-param.
Possibility for Local Adaptation	No	No	Yes	No
Model complexity depends on the # of data samples	No	No	Yes	No
Potential to create compact models irrespectively of data size and structure	High	Limited	Limited	High
Can final models provide insight into the problem, and increase system understanding	Yes	Hardly	Hardly	Yes
Complexity control possible	Yes	Yes	Yes	Yes
Danger to over-fit the data without explicit complexity control?	No	High	Limited	No
Danger of having insignificant variables in final models	Present	Present	Present	Not Present, or Heavily Reduced

Algorithm 1 This proposed algorithm for training a SR model: Initially we approximate the factors greedily using the automatic differentiation algorithm (7) and then fine-tune the factors until the convergence criterion is satisfied.

begin:

Set the generation counter to 0; and randomly initialize a population of individuals x_i.

Initialize the tuning parameter α. Set up the initial searching function.

Evaluate the fitness value for each individual in each iteration.

while stopping criteria are not satisfied **do**

for i=1 to population size **do**
start with selected initial searching function
for j=1 to randomly chosen index **do**

Mutation: generate a mutant individual using automatic differentiation strategy

Crossover: mix mutant individual and feasible individual to generate a trial offspring $\hat{x}_j$

evaluate the fitness value of offspring $\hat{x}_j$
if $\hat{x}_j$ is better than x_i **then**
$x_i = \hat{x}_j$
end if
end for
end for
end while

3.1. Problem Setup

To implement the SR-based genetic programming, the following steps are needed:

1. Binary representation conversion.
2. Initial population setup.
3. Fitness measurement.
4. Death, breeding, and mutation.

For this study, we choose six decimal digits of accuracy to create the possible solutions when converting the dataset using binary representations in step 1. Six decimal digits of accuracy corresponds to about 22 binary digits. A 22-digit binary string b can be converted to an integer k which is given by:

$$k = \sum_{i=1}^{22} b_i \cdot 2^{i-1} \tag{4}$$

The integer k is then converted to a real number u between 0 and 1 by:

$$u = k/(2^{22} - 1) \tag{5}$$

The number u is then converted to a real number r between -1 and +2 by:

$$r = -1 + 3 \cdot u \tag{6}$$

with these procedures, we have a mapping between genetic information b and the objective r.

3.2. Implementation Details

This section is mainly focused on demonstration of SR procedure, and for simplicity, during the modeling process, only a single input variable temperature (T) from the synthetic oil sands data is selected and the SR model output is the recovery rate (R). The complete simulation and analysis results considering more input variables will be described in detail in Section 4.

Each variable is initially given a string of 22 binary digits, which can be treated as an integer vector.

Choosing a population of $n = 50$ candidates, we create a 2 dimensional array of size 50×22.

A random initial population corresponding to T, which can be selected as an array of 0's and 1's, is given below as an example.

```
i   ----------b----------
----converted value of T----
#1   1000101110110101000111
=               0.637197
#2   0000001110000000010000
=               -0.958973
...
#50  1110000000111111000101
=               1.627888
```

The best individual is searched for the optimized output quantity $R(T)$.

In this study, the initial searching function is chosen as follows:

$$R(T) = T \cdot \sin(10\pi \cdot T) + 1 \tag{7}$$

Based on this initial selection, the iteration begins by evaluating the fitness of each candidate as follows:

```
i   ----------b----------
converted value of R(T)          Q(α)
#1   10001011101101010001 11
1.586345                  0.097567
#2   00000011100000000010000
0.078878                  0.124862
...
#50  11100000001111111000101
2.250650                  0.067429
```

Based on the fitness results, the following steps are performed for the next iteration:

- Out of the 50 candidates, remove 10 candidates with the lowest fitness;
- Let the 10 candidates with the highest fitness breed in pairs, creating 10 new candidates;
- Randomly select 2 of the nonbreeding, nondying candidates for mutation;

The size of the new population remains unchanged when it contains the best candidates from the previous population. The 10 candidates with lower score are replaced by new offspring of higher fitness score. For the intermediate 30 candidates, we modify two candidates randomly by mutation and keep 28 candidates unchanged.

For mutation, a candidate i can be picked to mutate. It is known that each candidate has 22 bits of genetic information. For example, one can pick index j between 1 and 22 and flip that bit:

```
b(i,j) = 1 - b(i,j)
```

For breeding, we assume parents i_1 and i_2 creating children i_3 and i_4. To achieve this, an index j between 1 and 21 is chosen and parental information can then be spliced together as follows:

```
b(i3,1:j) = b(i1,1:j)& b(i3,j+1:22)
b(i4,1:j) = b(i2,1:j)& b(i4,j+1:22)
```

Given the candidate #50 which is:

```
i   ----------b---------- converted value of R(T)

#50  11100000001111111000101      2.250650
```

If the fifth gene in this candidate is mutated, then the result is

```
11101000001111111000101   =>
-0.082257 => Q(α) = 0.736912
```

but if the 10th gene is mutated, then the following result is obtained:

```
11100000011111111000101   =>   2.343555
=> Q(α) = 0.065371
```

Thus, a mutation process can be controlled to improve the fitness value.

The following shows the results of breeding. For example, we make candidates #2 and #50 breed,

```
i   ----------b---------- converted value of R
#2   00000011100000000010000
0.078878          0.124862
#50  11100000001111111000101
2.250650          0.067429
```

The crossover point is defined as a selection of parental information to generate offspring. If the crossover point is after $j = 5$, the two children will be as follows.

```
----------b---------- converted value of R(T)
00000000001111111000101
0.940865            0.187592
11100011100000000010000
2.459245            0.078346
```

It is obvious that one child has a significant increase in the fitness function.

The GP procedure is terminated after 441 iterations. For the simulated data, intermediate results from the symbolic regression algorithm in several steps are shown in Table 2. The final results and the performance validation are included in Section 4.

Table 2. Results of SR descriptor via genetic programming for a single input variable T.

Iteration	$R(T)$
1	$T \cdot \sin(10\pi \cdot T) + 1$
6	T
8	$16.2 + T$
9	$34.3 + 1.08 \cdot T$
10	$61.7 + 29\sin(4.14 + 0.0519 \cdot T)$
12	$61.7 + 43.9\sin(4.23 + 0.0517 \cdot T)$
39	$18.3 + 73 \cdot logistic(0.465 \cdot T - 16.6)$
40	...
51	...
99	$18.2 + 0.000165 \cdot T^2 + 72.3 \cdot logistic(0.468 \cdot T - 16.7)$
120	...
441	$9.7 + 0.00025 \cdot T^3 + 3.75 \cdot T \cdot logistic(0.58 \cdot T - 18.39)$ $-0.054 \cdot T^2 \cdot logistic(0.58 \cdot T - 18.39)$

4. Results of SR Descriptor using Simulation and Laboratory Data

The oil sands data set contains three input variables: Temperature (T), pH, and Clay fines (Cf). The output is oil sands

processability index, bitumen recovery (R). Due to lack of industrial data, we use data from laboratory experiments together with the generated simulation data. The simulation data set contains 1000 data points, which are generated based on known and empirical knowledge about relations among variables. Furthermore, it is noted that symbolic regression requires repeated simulation from multiple data generation. This can be achieved by adding noises and changing noise profiles to repeat the simulation.

Applying the SR to the simulated oil sands dataset led to a descriptor (i.e. a mathematical expression) for the factor analysis of bitumen recovery after 742 iterations. In this section, we will construct and evaluate the performance of the descriptor by calculating the mean square error between the original data and reconstructed data for all input variables. In order to have comparable results, all of the temporary descriptor are set to the same termination criterion. We have set the maximum amount of iterations to 1000. As explained in Section 3, the final descriptor obtained by the SR for all three inputs is shown in Equation 8, which presents several sensible markers and their relationships constructed from the data used in this study.

$$\begin{aligned} R = {} & 0.39 \cdot T \cdot pH + 25.5 \cdot logistic(0.0597 \cdot pH) \\ & + 0.00783 \cdot pH^2 \cdot Cf^{-3} + 5.79 \cdot logistic(0.001 \cdot pH^3) \\ & + 0.049 \cdot T^2 - 0.035 \cdot T^3 - 0.002 \cdot T \cdot pH \cdot Cf^{-2} \\ & + 0.00369 \cdot Cf^{-4} \cdot logistic(0.001 \cdot Cf^{-3}) \\ & + ... \end{aligned} \tag{8}$$

It should be noted that the terms shown in Equation 8 are considered to be the key factors that can describe most of the relationship (higher coverage), and ... represents the other terms of lower importance that can be ignored due to lower coverage.

The SR descriptor shows a clear representation of the simulation data, which demonstrates characteristic features of input attributes. Such a descriptor can also be improved by integrating physical insights as constraints, such as function candidate blockers or generators during SR iterations.

Figure 2 shows the descriptor's output (i.e. the approximated recovery rate) with respect to temperature, in comparison with the training data (∘). The training dataset contains a mixture of real laboratory data and the simulation data, since the few number of laboratory data alone do not contain enough information to describe the bitumen recovery. It is clear that result in Figure 2 shows an excellent description of recovery dependence on temperature in Equation 8.

Figure 3 shows clearly how the bitumen recovery varies with slurry pH: starting from 10% at pH 4.5 with a rapid increase as pH increases to 7, followed by a slower increase to reach around 85% at pH 9.0. The results in Figure 4 show an opposite trend of decreasing biutmen recovery with increasing clays content.

Note that the descriptor found strongly depends on the selection of objective or fitness function. In this study, we simply select $Q(\alpha)$ (Equation 2) as the measure of the error when fitting the descriptor with the given input-output data. However, more

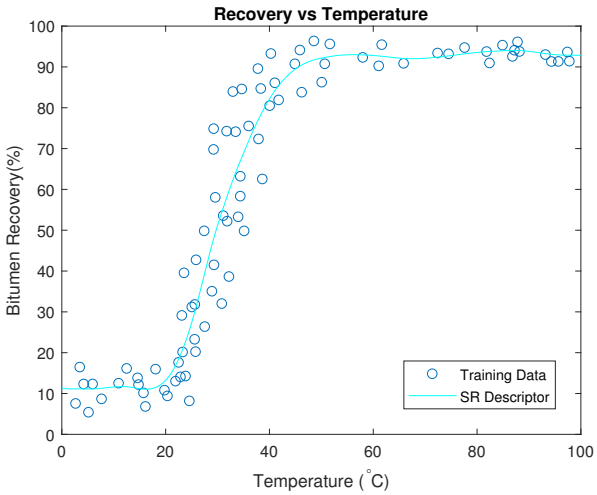

Figure 2: Symbolic regression prediction of the relationship between bitumen recovery and temperature.

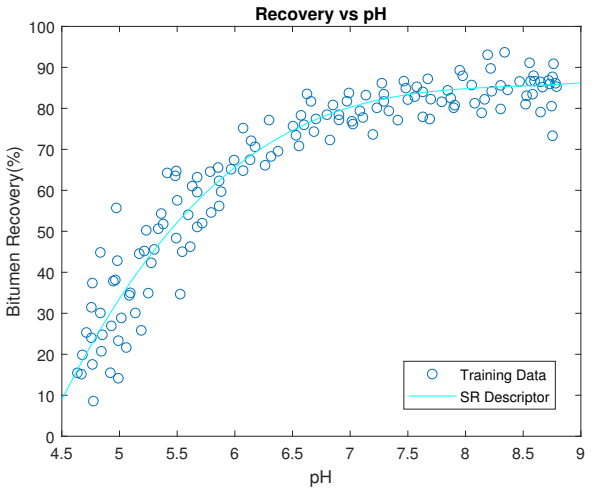

Figure 3: Symbolic regression prediction of the relationship between recovery and pH.

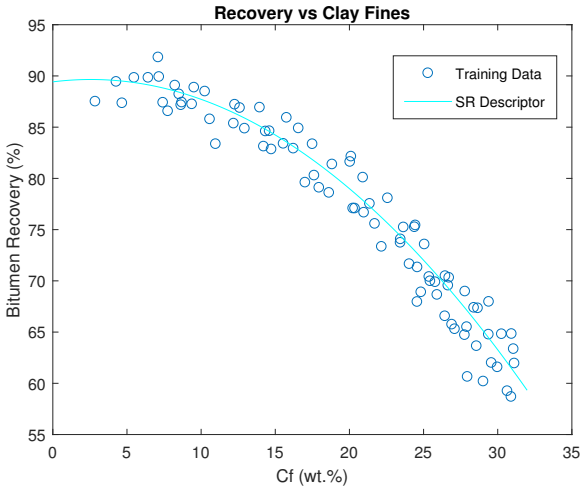

Figure 4: Symbolic regression prediction of the relationship between recovery and Cf.

selections can be tested such as stochastic universal sampling, or the tournament selection method (14, 2).

To assess the quality and accuracy of the descriptor, comparison of the predicted results from the descriptor and the original laboratory data is performed in Figures 5 to 7. Figure 5 shows the constructed relationship of the bitumen recovery with respect to the temperature, with comparison to the original laboratory data used in the dataset. By increasing the clay fines from 20 wt% to 30 wt%, two different curves are generated by SR descriptor which share the similar trend showing how the bitumen recovery is affected by temperature from 10 to 90 degree.

How the other two input attributes, such as pH value and the clay fines affect the bitumen recovery is also evaluated. Figure 6 and Figure 7 show the comparison of SR descriptor generated results with the original laboratory data. From these figures, it can be seen that the SR descriptor can reveal accurate trend and provides meaningful interpretation of the data.

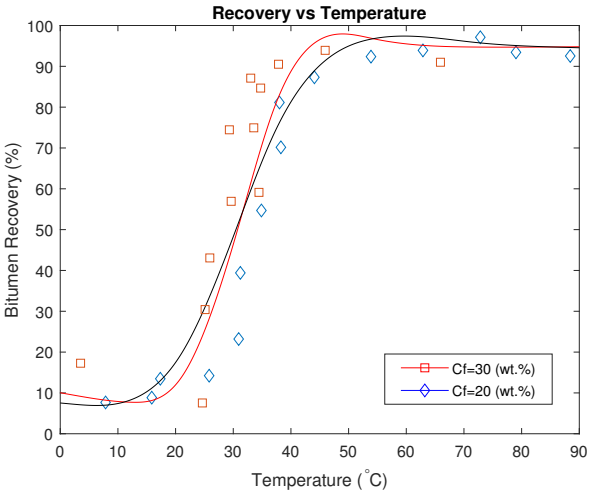

Figure 5: Symbolic regression prediction of the relationship between recovery and temperature. Blue hollow square indicate actual data points. By altering clay fines amount to 30%, red solid curve is visualized compared with original dark solid curve.

In Figure 8, the statistical performance of SR descriptor is further demonstrated, in which RMS errors between the SR result and the training and test dataset are compared respectively.

Finally, we evaluate the residuals between the observed value of the dependent variable and the predicted value (Figure 9). Both the training and testing data indicate a non random patterns and show a good fit for the SR model.

5. Experimental Results of SR Descriptor

5.1. Data Description and Results

The technology of near-infrared reflectance spectroscopy (NIRS) has been used for monitoring oil sands characteristics and operation procedure starting the 1950s. Quantitative prediction of oil sands composition such as bitumen content, water content and magnesium concentration can be conducted using various models such as principal component regression (PCR),

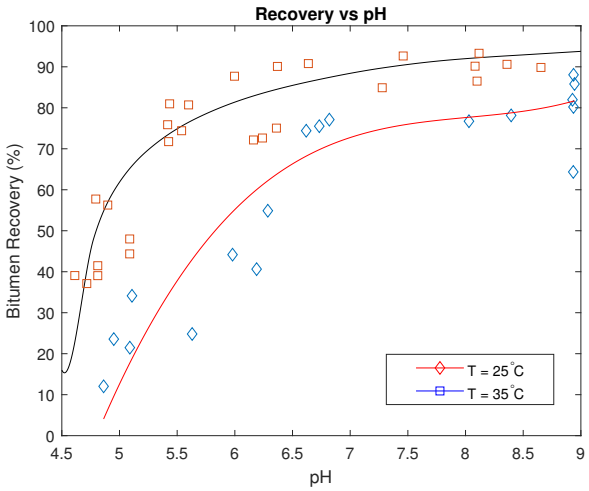

Figure 6: Symbolic regression prediction of the relationship between recovery and pH. Blue hollow square indicate actual data points. The difference between red and dark visualized curves is 10 degree alteration of temperature.

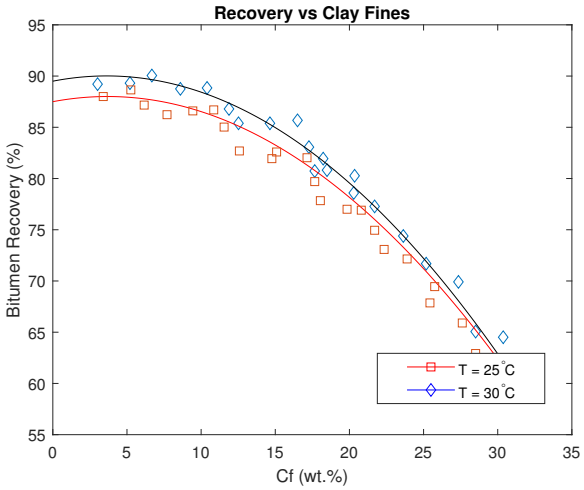

Figure 7: Symbolic regression prediction of the relationship between recovery and clay fines. Blue hollow square indicate actual data points. The difference between red and dark visualized curves is 5 degree alteration of temperature.

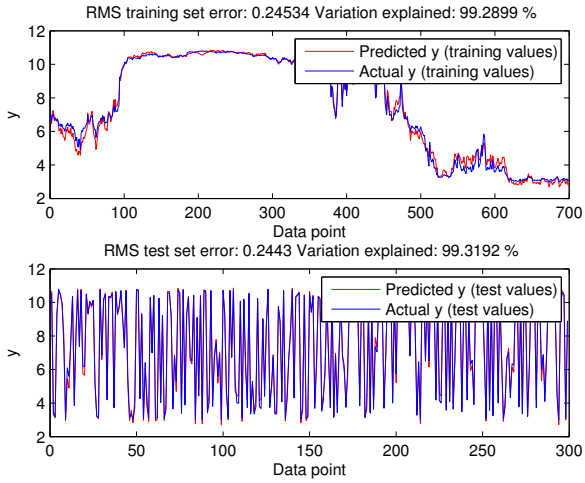

Figure 8: Symbolic regression model performance.

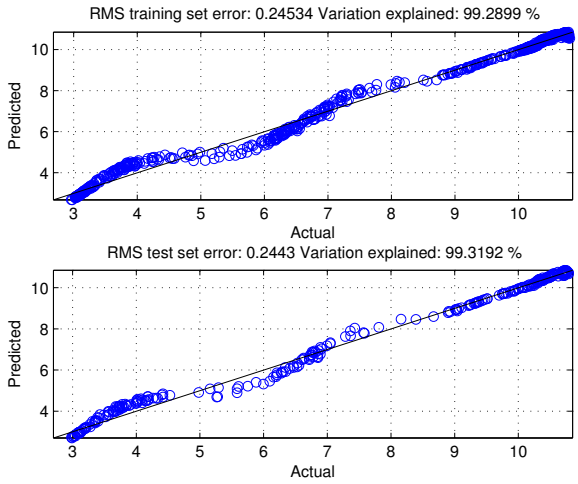

Figure 9: Symbolic regression prediction performance.

or artificial neural networks (ANN). The purpose of using NIRS data in oil sands analysis is to predict bitumen recovery. We will exhibit the SR descriptor using real world NIRS data in this section.

There are two probes for observing 9 measured variables: bitumen content, Ca^{2+} concentration, Clays content, fines content, water content, Mg^{2+} concentration, CL, Na and pH. At each probe there are 80000 data points. The NIRS data has been preprocessed for the purpose of converting spectrum data to numeric data.

Table 2. Symbolic regression setup

Setting	
Population size	10000
Function set	$+, -, x, /, sin, cos, tan, logistic$
Fitness function	RMSE
Selection method	Tournament selection
Crossover rate	90%
Mutation rate	5%
Number of generations	2000

As shown in Table 2, for the SR program setup, we select the initial operator set $(+, -, \times, \div)$ and the root mean square error (RMSE) as the fitness. The crossover rate is 90%, indicating that more offspring can be generated. The mutation rate is 5%.

We conducted the SR via GP and obtained the following SR model after 1487 iterations.

$$\begin{aligned} R = & \ sin(\frac{0.069 \cdot Bitumen}{0.0746 - Fines}) + logistic(0.0739 \cdot Ca^{2+} \cdot Mg^{2+}) \\ & - logistic(\frac{Na \cdot Bitumen - Ca^{2+}}{0.0368 \cdot Mg^{2+}}) \\ & - 0.00137 \cdot tanh(H_2O \cdot CL - Fines^{-1} \cdot Na) \\ & + logistic(\frac{Bitumen + 0.0239}{Fines - 0.764 \cdot H_2O})^2 \\ & + ... \end{aligned} \tag{9}$$

The above SR descriptor is obtained based on 8 input variables by removing the variable pH, because in the given NIRS dataset pH values do not vary much and are within a small range around certain constant.

Clearly the SR descriptor shown in Equation 9 is different from the one obtained from the simulation data in Section 4. In this NIRS dataset, the pH value is very stable while in simulated dataset its values are assumed to be varying. Moreover, more input variables with respect to more oil sands characteristics are included, such as magnesium concentration and clays, hence the pattern in NIRS data will exhibit different characteristics in the model.

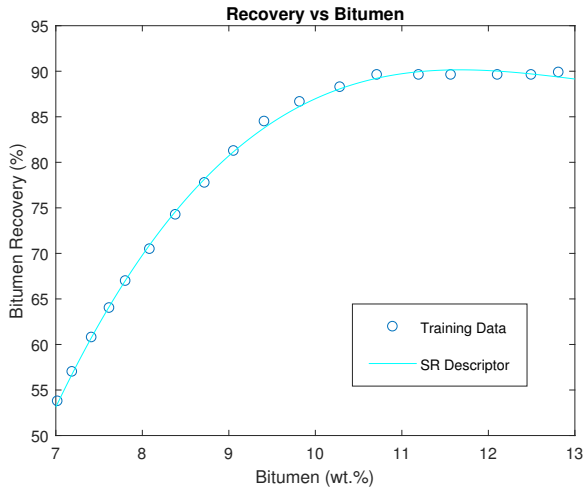

Figure 10: Symbolic regression prediction of the relationship between recovery and bitumen.

(You need to explain the new figures you added, e.g. Figure 11 -14 in the following paragraph...) The results showing relationship between several key variables and the bitumen recovery rate based on the SR descriptor are illustrated in figures 10 to 12. Figures 11 - 13, respectively. In Figure 11, ... In figure 12, ... To demonstrate the SR model prediction performance, Figure 14 illustrates how actual recorded values compare with the predicted ones on both training and test datasets. In this study, we choose the training dataset size as 500, and both of the test data size and the validation dataset size as 250. We use k-fold (k=5) method on the validation dataset. Higher density of overlapping between the data points and the line exhibits better accuracy of the SR model. For fitness results, the RMS error

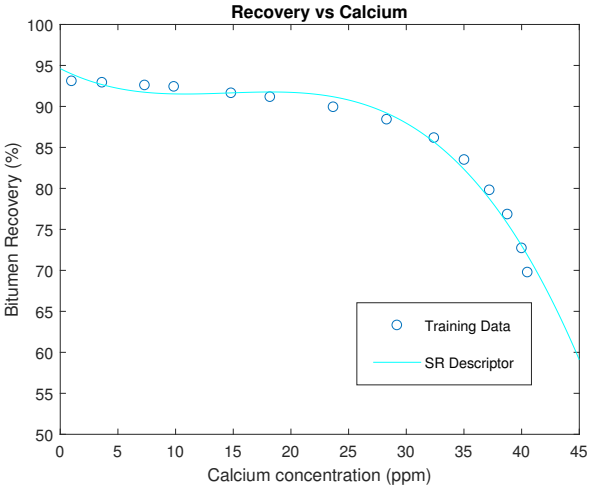

Figure 11: Symbolic regression prediction of the relationship between recovery and Calcium concentration.

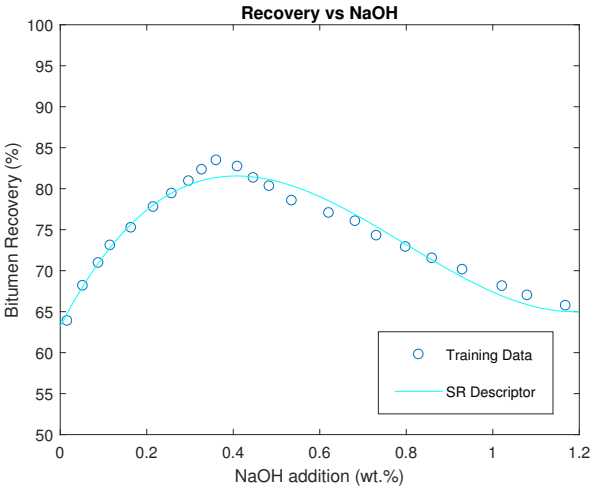

Figure 12: Symbolic regression prediction of the relationship between recovery and NaoH addition.

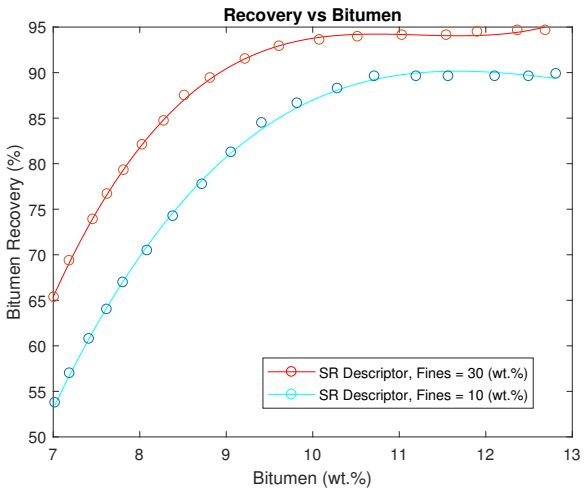

Figure 13: Symbolic regression prediction of the relationship between recovery and bitumen content by different fines condition.

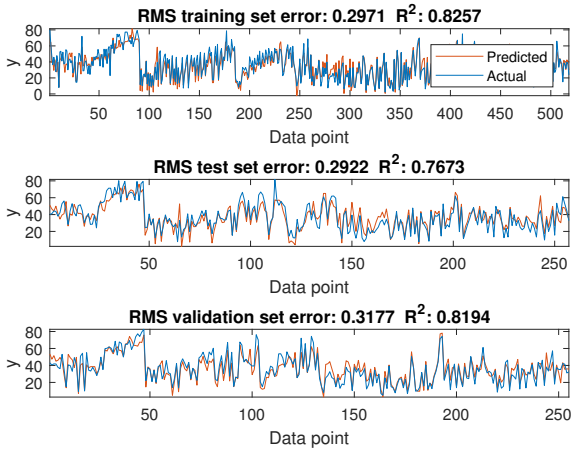

Figure 14: Symbolic regression model performance.

is 0.2971 on training data set and 0.2922 on testing data set. On validation data set, the RMS error is slightly lower than that of both training and test data. In this study, we also calculate R^2, the coefficient used for sensitivity analysis. The R^2 value is 0.8194 in the validation data and 0.7673 in the test dataset.

Figure 15 further shows the comparison between actual and predicted response. It shows a satisfactory match because the two responses align nicely along the diagonal line, which verifies the performance of the SR model. The vertical distance from the line to any point is the error of the prediction for that point. The RMS error is shown to be 0.3177 and R^2 error 0.8194, which are reasonably small.

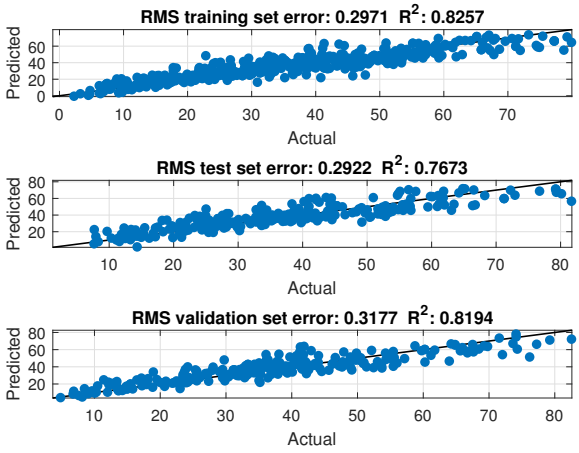

Figure 15: Symbolic regression prediction response.

5.2. *Discussions*

Remark 1 Physical Knowledge vs Model Accuracy

It is noted that there exists a trade-off between making prediction based on data and analysis based on physical knowledge. In many cases, knowledge about the process can help

improve the model by providing meaningful physical interpretation. However a key question is how to take advantage of physical knowledge. For example, in oil sands research, theoretical knowledge regarding bitumen recovery and processability of oil sands ores exist, but there still exist many uncertainties and unknown characteristics. Researchers rely on experimental data in this case. .... In this experiment, we compare results from SR model from different perspective which means prior knowledge is involved. Common facts from bitumen recovery processability can be hand-crafted into SR model as iteration blockers. For example, fines content has negative impact on bitumen recovery compared with bitumen content. Water content also has negative correlation with bitumen recovery. With those prior human knowledge, the following selected function sets :B/F,$1/Ca^2+$ have been structured as blockers.

Again, the ability of SR model to learn from experience means it can learn from experience: given enough experiment results regarding how to process bitumen, such as more oil sands characteristics and operation conditions, SR model can learn from this process and make accurate predictions.

We presented two results: R_1 without prior knowledge and R_2 with setup features.

$$
\begin{aligned}
R_1 = & \frac{-0.00337}{Fines - 5.79} + logistic(0.0158 \cdot Fines^{-1} \cdot Na) \\
& + logistic(0.0147 \cdot Ca^{2+} \cdot Bitumen^{1.1}) \\
& + 0.00762 \cdot tanh(\frac{Bitmen \cdot Clays^{-1.798}}{Fines^{2.6} \cdot CL}) \\
& - 0.00209 \cdot tanh(\frac{0.00186 \cdot Mg^2 + \cdot Bitumen}{logistic(0.276 \cdot Fines \cdot Ca^{2+})}) \\
& + 0.0122 logistic(CL \cdot H_2O) \\
& + ... \\
\\
R_2 = & sin(\frac{0.069 \cdot Bitumen}{0.0746 - Fines}) + logistic(0.0739 \cdot Ca^{2+} \cdot Mg^{2+}) \\
& - logistic(\frac{Na \cdot Bitumen - Ca^{2+}}{0.0368 \cdot Mg^{2+}}) \\
& - 0.00137 \cdot tanh(H_2O \cdot CL - Fines^{-1} \cdot Na) \\
& + logistic(\frac{Bitumen + 0.0239}{Fines - 0.764 \cdot H_2O})^2 \\
& + ...
\end{aligned}
\tag{10}
$$

Figure 16 shows the comparison between R_1 and R_2 results on bitumen contents. This comparison is purely for our objective reagaring statistically improvement. With prior knowledge, oil sands characteristics has been transformed into features that crafted into SR model. Moreover, SR model has been improved as a physical guideline for bitumen recovery processability.

Remark 2 Hybrid modeling

The advantage of this hybrid approach is that we can perform our data-driven model the physics of the problem by generating

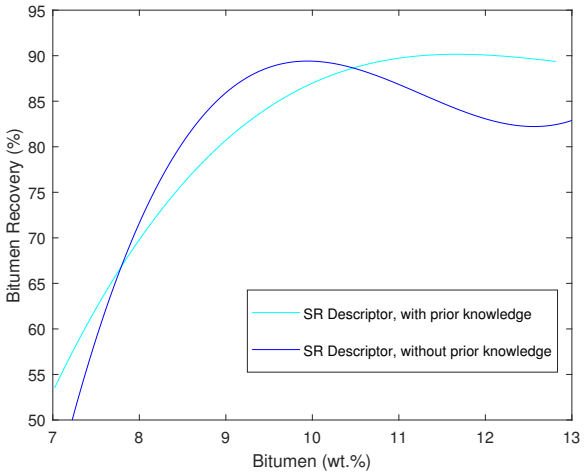

Figure 16: Symbolic regression prediction results with prior knowledge.

large amounts of training data from bitumen processability. The hybrid modeling methodology can be addressed in the algorithmic approach.

For traditional analytical modeling, basic rules or experiences only target a domain-specific formulation. Physical experiments are designed meticulously, intending with less human errors. However, there are may remaining modeling errors, for example due to lack of suitable models, or imprecise knowledge of extrapolation.

On the contrary, data-driven modeling such as symbolic regression approach can make an improvement of accuracy and most often address universal function approximators. In this process, feature selection of the training data, algorithm design and complexity are required.

We boldly propose a hybrid model that is a combination of parametrized analytical physical model with a data-driven model. The hybrid model requires a domain-specific model and also a data-driven model for the purpose of approximations. The results will have a gray-box characteristic combining explicit knowledge with a data-driven approach, which leaves a open discussion Figure 17.

6. Conclusion

We have introduced a novel symbolic regression model for oil sands recovery prediction, which is capable of selecting an optimum combination of function candidates and a set of mathematical operators so that the data can be described by the constructed mathematical descriptor. Furthermore, we show that the proposed technique is able to understand a combination of representations for oil sands recovery with respect to the sensible markers obtained using a simulation oil sands data and real world data set.

One important future work is to improve our experimental results and further interpretation for the industrial plant data.

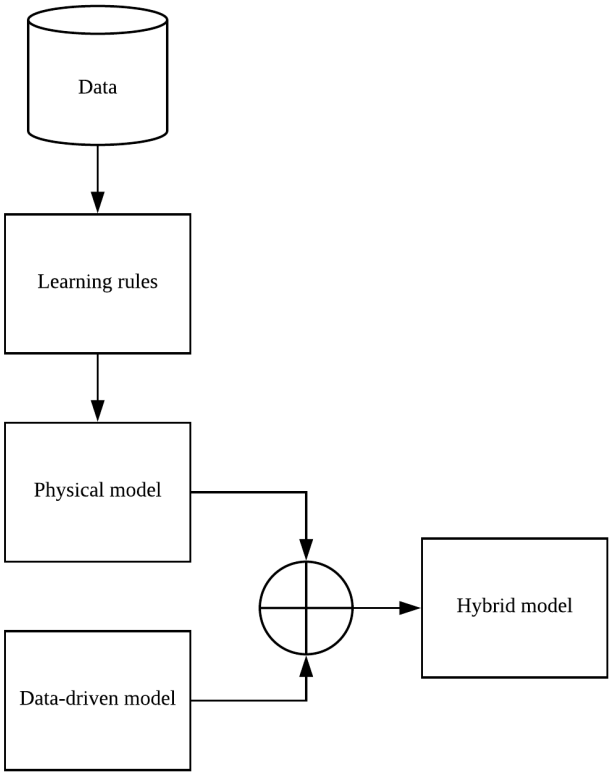

Figure 17: A overview of hybrid modeling.

Acknowledgments

This study was funded through Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] Berry, M. W., Browne, M., 2005. Email surveillance using non-negative matrix factorization. Computational & Mathematical Organization Theory 11 (3), 249–264.

[2] Blickle, T., Thiele, L., 1995. A mathematical analysis of tournament selection. In: ICGA. Citeseer, pp. 9–16.

[3] Fong, N., Ng, S., Chung, K. H., Tu, Y., Li, Z., Sparks, B. D., Kotlyar, L. S., 2004. A two level fractional factorial design to test the effect of oil sands composition and process water chemistry on bitumen recovery from model systems. The Canadian Journal of Chemical Engineering 82 (4), 782–793.

[4] Gen, M., Cheng, R., 2000. Genetic algorithms and engineering optimization. Vol. 7. John Wiley & Sons.

[5] Kumar, S. L., 2017. State of the art-intense review on artificial intelligence systems application in process planning and manufacturing. Engineering Applications of Artificial Intelligence 65, 294–329.

[6] Masliyah, J., Zhou, Z. J., Xu, Z., Czarnecki, J., Hamza, H., 2004. Understanding water-based bitumen extraction from athabasca oil sands. The Canadian Journal of Chemical Engineering 82 (4), 628–654.

[7] Neidinger, R. D., 2010. Introduction to automatic differentiation and matlab object-oriented programming. SIAM review 52 (3), 545–563.

[8] Sangdani, M., Tavakolpour-Saleh, A., Lotfavar, A., 2018. Genetic algorithm-based optimal computed torque control of a vision-based tracker robot: Simulation and experiment. Engineering Applications of Artificial Intelligence 67, 24–38.

[9] Schmidt, M., Lipson, H., 2009. Distilling free-form natural laws from experimental data. science 324 (5923), 81–85.

[10] Sushnigdha, G., Joshi, A., 2017. Evolutionary method based integrated guidance strategy for reentry vehicles. Engineering Applications of Artificial Intelligence.

[11] Tsoumakas, G., Katakis, I., 2007. Multi-label classification: An overview. International Journal of Data Warehousing and Mining (IJDWM) 3 (3), 1–13.

[12] Vladislavleva, K., Veeramachaneni, K., Burland, M., Parcon, J., O'Reilly, U.-M., 2010. Knowledge mining with genetic programming methods for variable selection in flavor design. In: Proceedings of the 12th annual conference on Genetic and evolutionary computation. ACM, pp. 941–948.

[13] Weninger, F., Schuller, B., 2012. Optimization and parallelization of monaural source separation algorithms in the openblissart toolkit. Journal of Signal Processing Systems 69 (3), 267–277.

[14] Whitley, D., 1994. A genetic algorithm tutorial. Statistics and computing 4 (2), 65–85.

[15] Zhang, Q. J., Sawatzky, R. P., Wallace, E. D., London, M. J., Stanley, S. J., 2004. Artificial neural network modelling of oil sands extraction processes. Journal of Environmental Engineering and Science 3 (S1), S99–S110.

Appendix A.

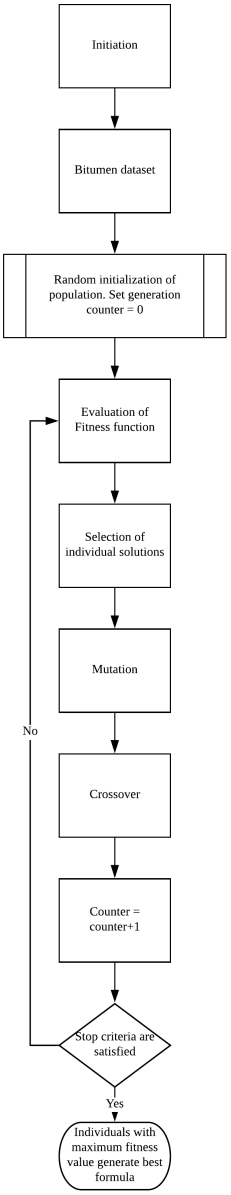

Figure A.18: Flowchart of symbolic regression algorithm.