

Reducing the Number of Test Cases for Performance Evaluation of Components

João W. Cangussu Kendra Cooper Eric Wong
Department of Computer Science
University of Texas at Dallas
Richardson-TX 75083-0688, USA
{cangussu,kcooper,ewong}@utdallas.edu

Abstract

Component-based software development techniques are being adopted to rapidly deploy complex, high quality systems. One of its aspects is the selection of components that realize the specified requirements. In addition to the functional requirements, the selection must be done taking into account some non-functional requirements such as performance, reliability, and usability, among others. Hence, data that characterize the non-functional behavior of the components are needed; a test set is needed to collect this data for each component under consideration. This set may be large, which results in a considerable increase in the cost of the development process. Here, a process is proposed to considerably reduce the number of test cases used in the performance evaluation of components. The process is based on sequential curve fittings from an incremental number of test cases until a minimal pre-specified residual error is achieved. The results from experiments with image compression components are a clear indication that a reduction in number of test cases can be achieved while maintaining reasonable accuracy when using the proposed approach.

1 Introduction

Component-based software engineering (CBSE) techniques hold the promise to support the timely, cost effective development of large-scale, complex systems; they are of keen interest to researchers and practitioners. However, there are numerous issues to address in CBSE including how to specify the functional and non-functional behavior of the components, how to evaluate, rank, and select components, how to predict the interoperability of components, etc.

A key issue in the specification of components is the problem of how to effectively test, or evaluate, the components in order to obtain quantitative data about their behavior. In particular, the non-functional behaviors such as

response time performance, memory usage, etc. need to be collected. Recognizing that complete sets of test cases are not possible and large, comprehensive sets of test cases can be prohibitively expensive, techniques to reduce the number of test cases while still providing meaningful information about the components are needed.

Here, we present an approach to select a reduced set of test cases that can be applied to a wide variety of components and non-functional properties. The approach is based on the use of polynomial curve fitting techniques, which are general approaches for representing a curve. The selection of an additional test case, which can be done either adaptively or randomly, is performed iteratively until an error tolerance is reached.

The approach is validated experimentally by selecting a reduced set of test cases for evaluating image compression components. These components implement well known algorithms including Arithmetic Encoding, Huffman, and the Burrows-Wheeler Transform. The non-functional behaviors under test are the compression time (how long does it take to compress a file) and compression ratio (how much smaller is the compressed file compared to the original file). Using our approach, as few as eight test cases are needed and selected in this experiment to capture the dominant behavior of the two non-functional behaviors. The selected test cases have a root mean square error of 0.0254 in comparison to the actual behavior of the component evaluated with a comprehensive set of 190 test cases. These results indicate the accuracy of the approach is excellent and offers a significant reduction in the number of test cases. The performance of the new approach is also experimentally evaluated, comparing the random selection of an additional test case with the adaptive approach. Our results indicate that non-linear behavior, e.g., compression ratio, has better performance with the adaptive approach; linear behavior, e.g., compression time, has better performance with a random approach.

The remainder of this paper is organized as follows. The new test case reduction approach is presented in Section 2;

the evaluation of the new approach is in Section 3. Related work is discussed in Section 4. Conclusions and Future work are presented in Section 5.

2 Proposed Approach

The goal of the proposed approach is to reduce the number of test cases needed to conduct a performance evaluation of non-functional behaviors of components. In general, a large number of test cases is needed to obtain a precise evaluation. The conjecture here is that a reasonably accurate evaluation can be achieved with a considerable reduction in the number of test cases. Also, the approach needs to be general; it should not be restricted for use on a specific subset of non-functional attributes

Any non-functional attribute of interest will always be a function of some input parameters, otherwise there is no need for testing. Therefore, the performance evaluation consists of finding the relationship between the input parameter(s) and the performance on the non-functional attribute. Both the input parameter (or some feature of the input parameter) and the non-functional attribute can be quantified and the relationship can be captured by some mathematical function. If the function is known in advance, then it can be directly used for the performance evaluation with no testing needed. However, this is rarely the case. In most scenarios, only the average performance is known which does not provide a comprehensive understanding of the behavior of non-functional attribute. In summary, the goal is to find the relationship using as few test cases as possible. Hereafter, the relationship between an input parameter x and a non-functional attribute y is referred to as $y = f(x)$.

Linear or non-linear regression models [10, 9] could be used to find the parameters of $f(x)$ if the general format of the function is known. For example, if y is known to have an exponential behavior such as ae^{-bx} regression models could be applied to find the values of a and b based on test cases (values of x) and the observed performance (values of y). However, in most cases, this relationship is not known and a more general approach needs to be used. Based on that, polynomial fit [5] has been chosen as, in general, any curve can be represented by a polynomial of a certain degree n , $p(x, n) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$. The problem is now the identification of the coefficients a_i , $i = 0, \dots, n$ of the polynomial. The *polyfit*(x, y, n) function available in MatLab [4] has been used to find the coefficients of $p(x)$ that fits the data points (x_i, y_i) , $i = 1, \dots, m$ in a least square sense. In the case of performance evaluation, m is the number of test cases and the pair (x_i, y_i) represents the input value and the corresponding observed performance. The method of least squares is based on the minimization of errors (least square errors); the distance between the actual point and the point in the fitting curve. The

best-fit curve of a given type is the curve that has the minimal sum of errors from a given data set. Suppose a sequence $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$ of m data points are given. The fitting function $f(x)$ has an error e associated with each data point, i.e., $e_1 = y_1 - f(x_1)$, $e_2 = y_2 - f(x_2)$, \dots , $e_n = y_n - f(x_m)$. Therefore, using the method of least squares, the best fitting curve has the property of minimizing Π in Eqn. 1.

$$\begin{aligned} \Pi &= d_1^2 + d_2^2 + \dots + d_n^2 = \sum_{i=1}^n d_i^2 \\ &= \sum_{i=1}^n [y_i - f(x_i)]^2 \end{aligned} \quad (1)$$

The pseudo-code in Figure 1 presents the steps of the proposed approach. Let us assume that a performance evaluation needs to be done within a pre-specified range a, \dots, b . For example, if image compression components are being considered, one may be interested in images with size ranging from 1K to 10G bytes of memory. The first step is then to select an initial small set of test cases to start with. In our experiments (refer to Section 3) the starting testing suite T is composed of three test cases $T = \{x_1 = a, x_2 = \frac{a+b}{2}, \text{ and } x_3 = b\}$. The stopping criterion of the approach is based on the comparison of two consecutive fits. That is, the curve Fit_1 is achieved using the results of test suite $T_k = \{x_1, x_2, \dots, x_k\}$, then a new curve Fit_2 is computed using the test suite $T_{k+1} = T_k \cup \{x_{k+1}\}$, where x_{k+1} is a new selected test case. Now, the root mean square error (RMSE), as given by Eqn. 2, between Fit_1 and Fit_2 is computed and the cycle stops when this error is less than a pre-specified threshold ϵ . In this case, the last computed test suite is the one that can capture the main behavior of the performance of the non-functional attribute under consideration.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=0}^N (x_1 - x_2)^2} \quad (2)$$

The behavior of the proposed solution is depicted in Figure 2. Let us assume $\epsilon = 10^{-2}$. Figure 2(a) shows the second iteration of the algorithm in Figure 1. In this case, the first test suite T_5 has four test cases a new point (test case) marked with a square is selected to compose T_6 . The value of the RMSE of the two fits is 0.9457 which is still larger than ϵ . After one more iteration, as seen in Figure 2(b), the RMSE between the fits using T_6 and T_7 results in an error of 0.0674 which means that more test cases are needed. The stopping criterion is reached in the next iteration (see Figure 2(c)) where the error between T_7 and T_8 goes to $0.0036 < \epsilon$. In this case a total of eight test cases is

```

01 begin
02   S = initial set of points and
03     associated outputs
04   error = infinity
05   while error > e
06     Fit1 = polyfit(S)
07     S = S + new selected point
08     Fit2 = polyfit(S)
09     error = RMSE(Fit1,Fit2)
10   end
11   TestSet = S
12 end

```

Figure 1. Test composition algorithm

needed to conduct the performance evaluation. As can be seen from Figure 2(d), the curve computed with the results of only eight test cases is very similar to the actual curve as indicated by $RMSE=0.0254$. That is, using only eight test cases the dominant behavior of the non-functional attribute has been properly captured.

In the example above, one aspect of the Algorithm in Figure 1 has not been considered: how to select the new test case in Line 06? Two approaches are considered in this paper. The first simply randomly selects the new test case from the available test cases within the specified range. The second approach does the selection in an adaptive way. The largest the gap between two consecutive inputs (assuming the test cases have been sorted), the less information the fitting method has to cover that area. Therefore, the selection approach is to fill this gap and increase the information used by the fitting method. After computing the gaps between each test case, the approach finds the largest gap and then try to find an available test case within this range. This step is needed because not all inputs in the original range from a to b may be available. For example, when testing the image compression with $a = 1K$ and $b = 10G$, images from all these sizes may not be available. Very large images may be hard to find and only 3 images may be available in the range from 5G to 10G. If no input is available, then the algorithm searches for the next largest interval. Notice that this selection approach will degrade to a full binary selection if all input values in the range are available; however, this is rarely the case.

3 Evaluation of the Proposed Approach

Three components for image compression are used here to evaluate the performance of the test case reduction technique described in Section 2. Although a large number of compression techniques exist, the decision to use Arithmetic Encoding (ARI), Huffman coding (HUF), and

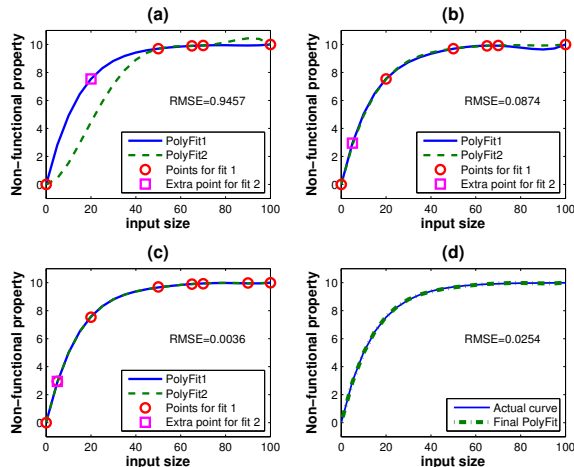


Figure 2. Results from applying the algorithm from Figure 1 to an exponentially shaped curve where (a) represents the results from iteration 2; (b) iteration 3; (c) iteration 4; and (d) is the comparison between the final curve and the actual one.

Burrows-Wheeler Transform (BWT) is based on their generality and availability. Also, two non-functional attributes are used in the examples: compression time and compression ratio. These two have been selected to represent a linear and a non-linear non-functional attribute. A large set of 190 images is available for the performance evaluation. The images range in size from 10 to 10M bytes. The images are not uniformly distributed. The non-uniform distribution is more realistic as smaller images (less than 1M byte) are more common and easier to find than larger images (more than 5M bytes). One image is used for each test case; the goal is to use the least number of test cases as possible while still capturing the dominant behavior of the non-functional attribute.

Figure 3 shows the results of applying both the adaptive approach and the random selection approach to the evaluation of compression time for the three components. The adaptive results are shown in Figures 3(a), (b), and (c) while the random selection results are shown in Figures 3(d), (e), and (f). A polynomial degree of 3 and an error of $\epsilon = 10^1$ have been used. The choice for a third degree polynomial is because it can represent a large variety of curves. Also, although the error may appear to be large at first, it is indeed small when compared to the values of the y axis (10^5 milli-seconds). As we can see in Figures 3(a), (b), and (c), the adaptive approach requires a total of 10, 23, and 15 test cases to capture the behavior of compression time for Huffman, BWT, and Arithmetic encoding, respectively. The observed behavior is linear and in the best case

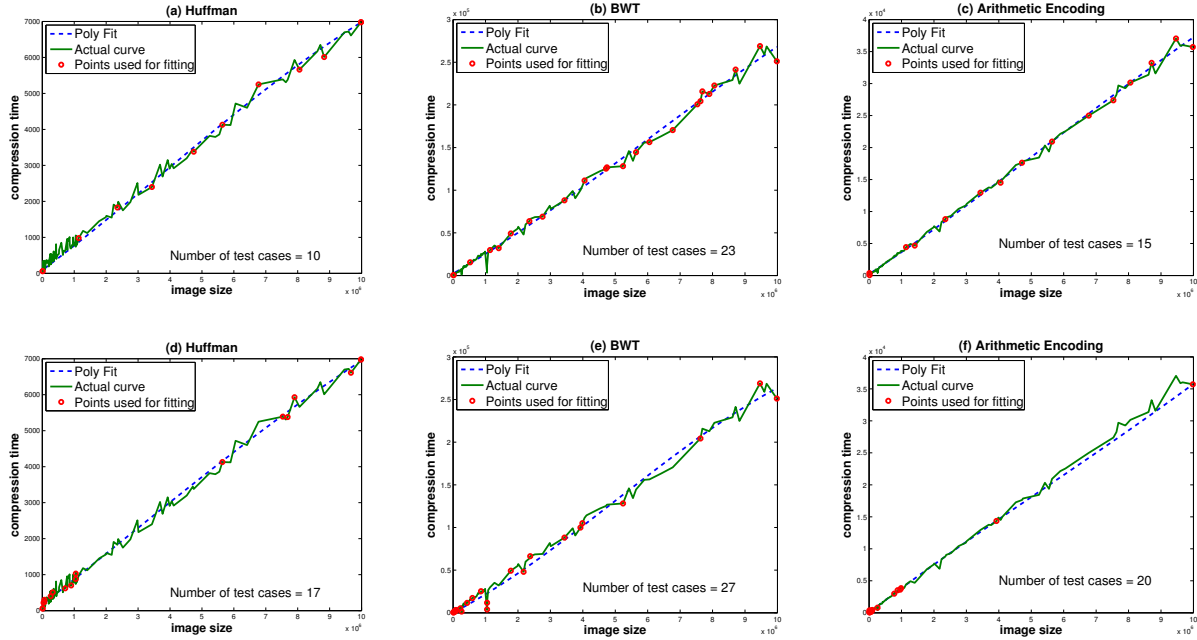


Figure 3. Results of the adaptive selection of test cases for compression time of: (a) Huffman, (b) BWT, and (c) Arithmetic Encoding. Results of the random selection of test cases for compression time of: (d) Huffman, (e) BWT, and (f) Arithmetic Encoding.

scenario only three points (test cases) would be needed to capture the behavior. However, the actual results are not a straight line and the use of only three points could lead to a different slope and possibly to a wrong characterization of compression time. In any case, the number of test cases needed seems reasonable small while appropriate to capture the behavior. The application of the random selection (Figures 3(d), (e), and (f)) resulted in, respectively, 17, 27, and 20 test cases needed for each of the components Huffman, BWT, and Arithmetic encoding. The results in this case are only slightly better for the adaptive selection when compared to the random selection. Only four extra test cases are required for BWF, seven are required for Huffman while the Arithmetic encoding requires five extra test cases. The accuracy of both approaches are almost the same with an average mean square error of 100 between the computed curve and the actual curve from all the available data points. Note again that a 100 units difference is small in the 10^5 scale.

As stated before, compression time for the evaluated components presents a linear behavior with respect to image size. To further evaluate the proposed approach the compression ratio, a non-functional attribute with a non-linear behavior, is analyzed next. In this case a polynomial of degree 3 and an error $\epsilon = 10^{-2}$ have been used. The error is smaller because the scale for the y-axis (compression ratio) is comparatively smaller. Figures 4(a),

(b), and (c) have present the results from the adaptive selection while Figures 4(d), (e), and (f) are the counterparts for the random selection of test cases. The adaptive selection required 12, 24, and 11 test cases. The results for the random selection present a considerable decline in performance. For the execution runs in Figures 4(d), (e), and (f), a total of 32, 83, and 31 test cases were required. Unlike compression time, the results in this case are much more favorable to the adaptive selection than to the random selection of test cases. Adaptive selection performs 2.6 times better than random selection for the Huffman component. The improvements for BWT and Arithmetic encoding are, respectively, 3.4 and 2.8.

When considering compression ratio, the adaptive approach has performed considerably better than the random approach. This is due to the non-uniform distribution of the image sizes and the non-linearity of the requirement under consideration. In general, when test cases are uniformly distributed, the adaptive and the random selection approaches will have a similar behavior. The adaptive approach tries to fill the largest interval between two input values, which tends to make the selected inputs uniformly distributed. Therefore, if the inputs are already uniformly distributed, it is expected that random selection will behave in a similar manner. However, this is not the case for the compression ratio and since images are clustered, the selection of a new image may lead to a large difference between

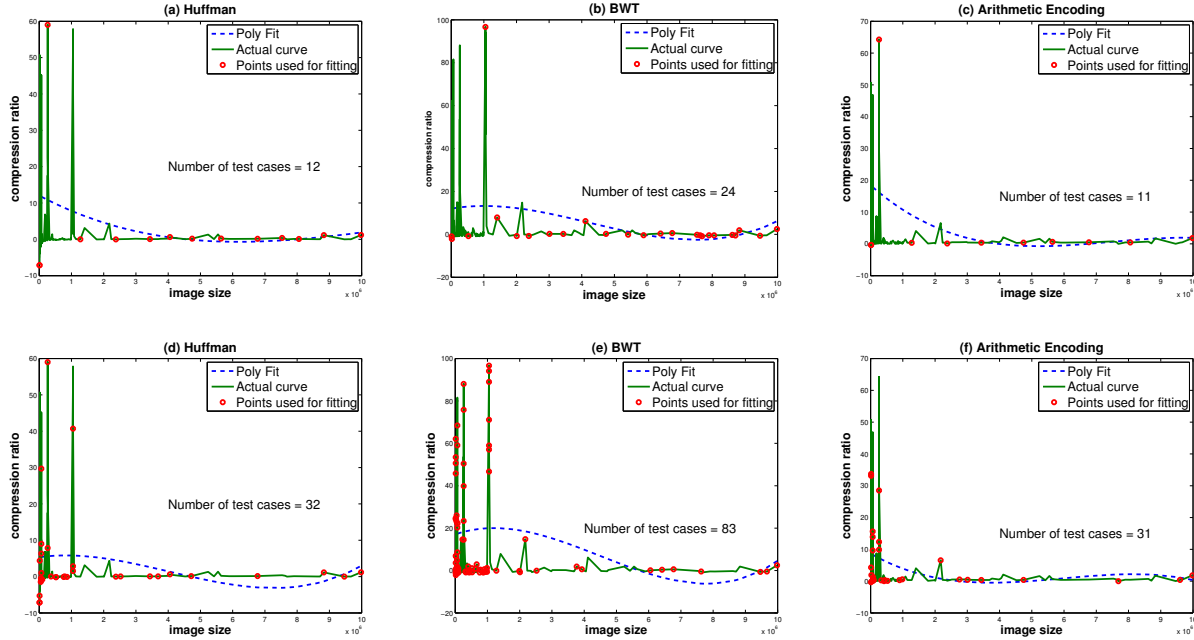


Figure 4. Results of the adaptive selection of test cases for compression ratio of: (a) Huffman, (b) BWT, and (c) Arithmetic Encoding. Results of the random selection of test cases for compression ratio of: (d) Huffman, (e) BWT, and (f) Arithmetic Encoding.

the two polynomial fits and consequently a larger number of test cases will be required. The expectation is that the more complex the behavior of the requirement (meaning, the more complex the curve to fit), the larger the number of test cases (points) needed to capture its behavior. The non-linearity of compression ratio is an indication of this complexity leading to a larger number of test cases required to capture its behavior.

The results presented in this Section provide a good indication that the new test reduction approach can select a small number of test cases to conduct an accurate performance evaluation of components. However, more extensive evaluation of the proposed approach needs to be conducted to further verify this conjecture. This has been deferred to future work. Another aspect that still needs additional evaluation is the impact of the values of the degree of the polynomial and the value of the error ϵ on the performance of the approach. The conjecture is that the higher the degree of the polynomial and the smaller the value of ϵ the larger the number of test cases required to capture the behavior. Once this conjecture is confirmed (with future case studies and simulation runs) the next step would be the optimization of these values. That is, given a specific non-functional attribute, a certain degree for the polynomial and a certain value for ϵ should be found to minimize the number of test cases required while maximizing the accuracy of the fit. This problem is also deferred to future work.

4 Related Work

To the best of our knowledge, we are not aware of any studies with a similar objective as what is reported here. The rest of this section focuses on work in three categories: test cost reduction, adaptive testing, and component-based testing.

Test Cost Reduction: Marre et al. [8] used a spanning set of entities to generate test suites in order to estimate and to reduce the cost of testing based on the observation that one test case generally covers more than one entity. As a result, if we can identify “a subset of entities with the property that any set of tests covering this subset covers every entity in the program,” then we can reduce the cost of testing. They presented a method for finding a minimum set of entities of full coverage and an automated method for finding the corresponding spanning set.

Ling et al. [7] proposed a decision-tree learning algorithm to build more effective decision trees in order to minimize the sum of the misclassification cost and the test cost. Their algorithm is based on cost-sensitive learning methods such as a Markov Decision Process. They also explained the problems of other approaches and claimed that their algorithm is superior to the others.

Adaptive Testing: Adaptive Random Testing is an active research topic because of its effectiveness under some well

distributed input domains [1] The actual results depend on the “distances” between different test values. However, such distances have only been defined for integers and other elementary values. Ciupa et al. [2] extended this idea by introducing an “object distance” to test object-oriented programs. A Distance-Based Adaptive Random Testing (D-ART) method was also proposed based on object distance.

Component-based Testing: Damm et al. [3] proposed a framework for automated component testing. This approach is based on Test-Driven Development (TDD) which creates the test cases before developing software components. The proposed framework extends the traditional TDD (which is for each class and method) to the component level, which needs to test for each component interface. As a result, defects can be detected earlier in the development cycle to reduce the overall cost of testing and debugging.

One difficulty of testing software components among others is testing them under numerous hardware, operating systems, and third-party COTS components. Grundy et al. [6] proposed an approach using a “validation agent” and a “component aspect” to resolve this problem. They are used at the deployment time to validate the components. The validation agent tests functional and non-functional aspects of software components in an actual deployment situation, whereas the component aspect cross-cuts the aspects of the components to increase its usability.

5 Conclusions and Future Work

A new approach that selects a reduced set of test cases for the accurate performance evaluation of components has been presented in this work. The approach is based on the use of polynomial curve fitting techniques. The approach begins by using a small set of initial test cases. Using an iterative approach, a new test case is found and added to the test case set. The new test case may be selected either randomly or using an adaptive technique. The test cases are executed; the performance evaluation results are used to compute a new curve. When the RMSE between the previous and the current curves are below a threshold, then the reduced test case set has been found.

The approach has been experimentally validated using a set of components that implement well-known image compression algorithms. Using our approach, as few as eight test cases are needed and selected in this experiment to capture the dominant behavior of the two non-functional behaviors. These results indicate the accuracy of the approach is excellent and offer a significant reduction in the number of test cases.

The performance of the new approach has also been experimentally evaluated, comparing the random selection of an additional test case with the adaptive approach. Our re-

sults indicate that non-linear behavior, e.g., compression ratio, has better performance with the adaptive approach; linear behavior, e.g., compression time, has better performance with a random approach.

There are several interesting directions for future work. The first is to apply the approach to additional sets of components and evaluate different non-functional attributes, to improve the validation of the approach. The second is to investigate the impact of the values of 1) the degree of the polynomial used in the curve fitting calculation and 2) the error threshold. When the impact of these values are thoroughly quantified, then the values for non-functional attributes could be optimized.

References

- [1] T. Y. Chen, H. Leung, and I. K. Mak. Adaptive random testing. In *Lecture Notes in Computer Science*, 3321:320-329, 2004.
- [2] I. Ciupa, A. Leitner, M. Oriol, and B. Meyer. Object distance and its application to adaptive random testing of object-oriented programs. In *Proceedings of the 1st International Workshop on Random Testing*, 55-63 July 2006.
- [3] L. Damm and L. Lundberg. Results from introducing component-level test automation and test-driven development. *Journal of Systems and Software*, 2006.
- [4] Walter Gander, J. H. Masaryk, and J Hrebicek. *Solving Problems in Scientific Computing Using Maple and MATLAB*. Springer-Verlag, 1997.
- [5] Walter Gautschi. *Numerical Analysis: an introduction*. Birkhauser Boston, Cambridge, MA, USA, 1997.
- [6] J. Grundy, G. Ding, and J. Hosking. Deployed software component testing using dynamic validation agents. *Journal of Systems and Software*, 2005.
- [7] C. X. Ling, Q. Yang, J. Wang, and S. Zhang. Decision trees with minimal costs. In *Proceedings of the 21st International Conference on Machine Learning*, 69-76 July 2004.
- [8] M. Marre and A. Bertolino. Reducing and estimating the cost of test coverage criteria. In *Proceedings of the 18th International Conference on Software Engineering*, 486-494 May 1996.
- [9] G. A.F Seber and C. J. Wild. *Nonlinear Regression*. John Wiley & Sons, Inc., 2006.
- [10] George A. F. Seber and Alan J. Lee. *Linear Regression Analysis*. Wiley-Interscience, second edition edition, 2003.