Reviewer 1

One weakness I see in the evaluation is in the choice of case studies. More specifically, a very common case in practice which is highly relevant to the suggested framework is that of a test space that contains many MFS of a low degree. Such a case is not represented in the evaluation. **When there are many MFS of a low degree, there are many failing test cases, hence many calls to the identification component.** In addition, the probability that a newly generated test case will contain multiple MFS that will cause an identification failure is significantly higher. As illustrated by Figure 2, such a test case will be skipped without contributing anything to the accumulated coverage, and a new test case will be generated instead. **The question whether under these conditions the framework still achieves a reduction in the number of multiple MFS and in the total number of test cases should be evaluated.**  
  
One more topic I am missing is addressing **test space constraints.** Since almost all real-world industrial CT models contain constraints, the question how do they impact the suggested framework is of significant importance. For example, an implicit schema can stem not from the interaction of an MFS with other MFS but rather from the interaction between the MFS and the model constraints. How do the authors account for such lost schemas? It is also not mentioned whether the 5 case studies contain constraints. Incorporating case studies with constraints increases the validity of the evaluation results.

Additional comments:  
- 4.2 before EQ5: the text mentions that the mutation should not include the "currently identified MFS". But the MFS is not identified yet - is the intention here the candidate MFS? This should be explained more clearly.

- 5.3.2 masking effects: I did not understand the possible explanation suggested for the lack of gap between ict and sct in terms of masking effects. Was the intention that since sct covers the same schemas more times, the chances of them reappearing in passing tests hence reducing masking is higher? Please clarify.

- Potential future work: incorporate bug fixing information into the framework, i.e., MFS combinations becoming non-MFS - how can the framework utilize this information in an efficient and effective way?

- Reference number 23 is missing the author names

- The paper can benefit from proof reading as it contains numerous typos and grammar mistakes

Reviewer 2

**The motivation** of covering all t-wise interactions before moving to the debugging phase is not clear. Why not generate tests while identifying failure-inducing interactions then debug and fix the problems and then re-test by augmenting the test suites. This will probably result in reduced test suites.  
  
The idea behind the approach is **quite similar** with what is proposed in code-based fault localization, e.g., Jeremias Rößler, Gordon Fraser, Andreas Zeller, Alessandro Orso: Isolating failure causes through test case generation. ISSTA 2012: 309-319. Although different in context, I think the paper will benefit by discussing it since it relies on the same idea. Similarly, the idea of generating CT tests by selected dissimilar tests and prioritizing them at the same time is related with the similarity t-wise test selection used in product lines, i.e., Christopher Henard, Mike Papadakis, Gilles Perrouin, Jacques Klein, Patrick Heymans, Yves Le Traon: Bypassing the Combinatorial Explosion: Using Similarity to Generate and Prioritize T-Wise Test Configurations for Software Product Lines. IEEE Trans. Software Eng. 40(7): 650-670 (2014). Additionally, the paper states, “in the generation stage, testers have no knowledge of the possible MFS, and surely it has opportunities that multiple MFS appear in the same test case”. It seems that dissimilar tests such as those produced by the above paper are the most appropriate to handle such cases.  
  
**I also think that a short discussion of combinatorial test generation approaches and code-based fault localization should be given, in the related work section.**  
In the approach, why it is mandatory to forbid MFS when augmenting the test suites? It is possible that MFS can interact with input parts, not exercised by the employed test suite, and hide the fault. In other words, why is it assumed that every time that a specific failure-inducing schema is used it triggers the failures. This might not happen under higher strengths. I wonder whether this was observed in the conducted experiments.  
  
**I missed a discussion on how the input constraints are handled? Additionally, why higher t-wise strengths are not always resulting an improved precision**? An explanation should be given.  
  
Why in TCAS the approach is not as good as in the other subjects? Is it because in TCAS the input combinations do not always trigger the mutants? I think that an explanation about this is important as it might indicate limitations of the proposed approach. Additionally, when discussing the results of TCAS the paper states, “**Under this condition, both approaches will be transferred to a normal covering array**”. Please revise the sentence, as it is unclear.  
  
**I think that the paper does a pretty good job in evaluating its propositions on real world subjects. However, I believe that the employed subjects, tests and models should be available in the companion website of the paper. This will enable replication and will help researchers validate their CT approaches on these subjects. Additionally, the manual identification of the MFS introduces a validity threat, which can be reduced by making these data available.**  
  
I also think that the paper can benefit by adding some new results. These involve the performance of the examined approaches and the test size of a CT test suite that ignores the masking effects. The former will indicate the performance impact of the proposed approach on CT test generation and the latter the impact of masking on the test suite size.

Reviewer: 3  
  
  
First of all, it is not clear whether the ultimate goal of the proposed approach is to identify failure-inducing option setting combinations or to obtain full coverage under the tested t-way coverage criterion or both. The authors should make this clear in the paper.  
  
Regardless of the ultimate goal, one major concern is the contribution of this work. Using a greedy, one-configuration-at-a-time approach to compute covering arrays, changing one option setting at a time (OFOT) for fault localization, expressing failure inducing combinations as constraints to avoid previously known failing sub-spaces, and feedback-driven, adaptive CIT are not new ideas at all. The proposed approach simply combines OFOT with covering array generation in a rather trivial way, such that likely failing sub-spaces are avoided and that previously covered combinations are not required to be covered repeatedly.  
  
**Another issue is that throughout the paper it is claimed that the proposed approach determines minimal failure-causing schema (MFS) as it is defined in Definition 4. However, as also discussed in Section 4.5, the proposed approach does not guarantee to find MFS. Therefore, either the definition or the terminology used throughout the paper should be changed, because the proposed approach can in general determine \*\*likely\*\* failure-inducing option setting combinations, nothing more.**  
It is good that the authors summarized the shortcomings of traditional CIT approaches in the presence of masking effects and multiple MFS in a single configuration. However, it is not clear how masking effects differ from multiple MFS. I would say that multiple MFS may cause masking effects. If so, these are not two separate concepts (in the sense that one causes the other) and they should be treated accordingly in the paper.  
  
On a related note, dealing with masking effects/multiple MFS is crucial for the proposed approach, as both the effectiveness and the efficiency of the proposed approach can greatly suffer in the presence of them (as also noted by the authors). The paper claims that the proposed approach can deal with multiple MFS, thus masking effects, in a single configuration.  However, it turns out that this is due to a heuristic, which aims to reduce the likelihood of having multiple MFS in a single configuration, which in turn is due to the way the proposed approach operates, i.e., one failure at a time. **That is, the proposed approach does not guarantee to avoid all multiple MFS/masking effects, as was also observed in the experiments**. For example, in the example given in Figure 5, which is used to illustrate the proposed approach, if testing started with configuration “0001” instead of “0000”, none of the failure inducing combinations would have been found, thus the full coverage under tested t-way coverage criterion would not have been obtained. Considering the current level of contribution of the paper, developing approaches for resolving multiple MFS/masking effects once they surface themselves, can greatly improve the contribution of the paper.  
  
In the experiments, the proposed approach is compared to FDA-CIT – a feedback driven, adaptive CIT process. However, there are several issues that need to be addressed with these experiments:  
  
**First, the proposed approach assumes that all failures are deterministic, which should be mentioned and discussed early in the paper, as this greatly reduces the practicality of the approach.**  
Second, it is not clear how the first configuration to start the proposed approach was chosen. This is important because the performance of the proposed approach depends on the first configuration (especially in the presence of multiple MFS).  
  
**Third, the number of configurations required by the identification part of the proposed approach grows linearly with the number of configuration options**. However, in the experiment the maximum number of options used was 13, which is quite small. For example, the size of a 2-way covering array created for 10 binary options, can be 6, whereas that created for 6435 binary options can be 16. That means that while the proposed approach will require 10 additional test cases for locating a single MFS in the first case, it will require 6435 additional test cases for the same failure in the second case. On the other hand, FDA-CIT will use 16 rather than 10 test cases to determine the likely failure causes, as FDA-CIT does not require additional test cases for identification. When this coupled with the fact that most of the test cases required by the proposed approach were used by the identification part (Table 8), it necessitates that, to perform a fair comparison, the empirical studies reported in the paper should be repeated by systematically increasing the number of configuration options.  
  
**Fourth, the proposed approach assumes that only one test case is used for testing.** Here, I distinguish between configurations and actual test cases used in these configurations to test the system under test. For example, what if you have hundreds of test cases to run in each of the selected configurations. Note that each test case can have different failing patterns. It seems like the proposed process should be carried out separately for each test case, as it may not be safe to invalidate a failure-inducing combination discovered for a particular test case when running other test cases. If so, the number of additional configurations needed will grow linearly with the number of configuration options times the number of failing test cases. Therefore, for a fair comparison, the performance of the proposed approach should be compared to that of FDA-CIT in the presence of multiple test cases.  
  
Fifth, it is strange to see that while the F-measures obtained from FDA-CIT were so low (Table 13), the tested t-way coverage measures for FDA-CIT were similar or better than those obtained from the proposed approach (Table 14), especially for large values of t, e.g., t=3 and t=4. Could this be because of the way the F-measures were computed? Seems like automatically identified failure-inducing combinations were symbolically compared with actual combinations, which may be misleading. For example, FDA-CIT can determine a portion of the actual failure-inducing combination at each iteration. Therefore, all the portions related to the same failure should be combined before any performing any comparison. Furthermore, in FDA-CIT, superfluous options can crept into the classification models to protect the integrity of the models, for example to ensure that the classification tree has a single root. Therefore, it may make more sense to compute precision, recall, and f-measures in terms of the correctly/incorrectly identified failure causing schemas of degree t, rather than symbolically comparing the option setting combinations.  
  
Furthermore, it is not clear how the faulty versions of the subject applications used in the experiments were chosen. For example, only one faulty version marked as #55905 seems to have been chosen for Tomcat (Table 6). Why and how?  
  
Section 4.2 can greatly be shortened as it simply describes a greedy, one-configuration-at-a-time covering array construction approach. The equations introduced in this section do not really help, as they are not used in the remainder of the paper.  
  
Section 2.1: For a better taxonomy of construction methods for covering arrays, the author should refer to Nie et al.’s survey (ref [38] in the paper.)  
  
Author names are missing from references [15] and [23].  
  
Line 26, second column, page 4: a space character is needed before parenthesis.  
  
Line 34, first column, page 6: “a validate schema” -> “a valid schema”  
  
Line 10, second column, page 6: “that was” -> “that were”  
  
Line 46, second column, page 14: “One the other hand” -> “On the other hand”