REPLY TO REVIEWERS

Comments from the Associate Editor

Dear Prof. Cleland-Huang,

Please thank your reviewers for their excellent review notes from our prior draft. Using that feedback, we have extensively revised the prior draft which we now submit for your consideration.

In the notes below, we detail how this draft differs from the first one. Note that paragraphs in ***bold italics*** are text taken from the prior review.

You write: ***Based on the reviewers comments I am unable to recommend this paper for TSE. The recurring theme in the reviewers comments (both public and private) is that the paper is poorly written and therefore hard to understand.***

We concur. This new paper is built around a structure proposed by Reviewer2 and Reviewer3 who both said that the core issues of this paper are:

- **RQ1 (speed):** Does GALE terminate faster than other SBSE tools?
- **RQ2 (quality):** Does GALE return similar, or better, solutions than other SBSE tools?

Please note that we now list these questions in the introduction and that our new results section is based on these two questions.

As to the half dozen research questions in the last draft, we have replaced them with the above two.

We have also improved the last draft with structural features and have added much more detail in order to improve its comprehensibility:

1) We offer a (very) abridged summary of the new algorithm, on page 2;
2) We added extensive notes on the background of search-based SE on page 2, page 3;
3) We offered detailed pseudo-code of the new algorithm on page 4, page 5, page 6;
4) We rewrote the results section on page 8, page 9, page 10 to greatly simplify the results section.
5) We added exact details on the algorithm's pseudocode on page 4, page 5, and page 6.

You also write: ***It lacks sufficient details of the approach. One reviewer (#1) has more serious issues with (a) the scope of the paper and (b) the delta of the contribution above and beyond prior work.***

We completely agree that the last draft of the paper had the wrong scope: it should have been more about the GALE algorithm and less about the agile model. That has been fixed in this draft (using the excellent suggestions from the last round of review regarding the two core research questions, shown above).

We further agree that the new version of POM3 is not unique or interesting enough to justify journal publication at TSE. Instead, this paper is about a new search-based SE algorithm that runs orders of a magnitude faster than the state-of-the-art.

Also, to increase the delta from prior work, we have added a completely new case study (the NASA human-automation cockpit model called CDA- see pages 7 through page 9).

Reviewer 1

The authors build upon the previously published POM1 and POM2 models and propose GALE to improve the performance of POM3 (specifically, to reduce the number of evaluations that are needed). The authors base the experiments on four scenarios and show that the proposed method achieves results that are comparable to other methods (e.g., NSGAII) but with fewer number of model evaluations. One major problem of the current manuscript is that how much contribution POM3 makes.

We very much agree with you. In this draft, the contribution of this paper is the GALE algorithm.

As to the POM3 model of agile development, that is now merely just one of the eight models studied in this paper. Using those models we make the case that our "geometric active learner" can find competitive solutions for SBSE problems, using 10 to 100 times fewer evaluations than established practice.

Another major drawback relates to the four scenarios that the author used to evaluate GALE. The choices were made without any explanation and justification – except that "it is an interesting subset". However, many other scenarios may be equally "interesting" and POM3d (small, seldom changing projects) does *NOT* seem to belong to agile development's scope of applicability and therefore is *NOT* interesting.

We completely agree- especially about POM3d (which has been excluded from this draft).

As to why we still use the other POM3 scenarios; it turns out that there was an interesting meta-level issue with how the other MOEAs handled the POM3 model. When exploring that issue, we found it useful to run three different input scenarios (in order to verify that the issue was not a quirk of some input). We now say at end of section 4.1.2 that:

> When we ran POM3 through various MOEAs, we noticed a strange pattern in the results (discussed below). To check if that pattern was a function of the model or the MOEAs, we ran POM3 for the three different kinds of projects shown in Figure 10. We make no claim that these three classes represent the space of all possible projects. Rather, we just comment that for more than one class of agile projects, GALE appears to out-perform NSGA-II and SPEA2.

But apart from that, we acknowledge your point that these scenarios do not reflect the space of all possible agile projects, and merely that GALE can work for the scenarios (arbitrarily) selected.

A significant flaw in formulating research questions appears between RQ3 and RQ4. The authors have

presumed the answer to RQ1-RQ3 is "yes". A more empirically rigorous way is to formulate null hypothesis and formulate further hypothesis based on the *actual* statistical test results.

We agree that the research questions in the prior draft were poorly formed. We do not reuse them here. Instead, we base this paper on two core issues identified by reviewers in the last round:

- **RQ1 (speed):** Does GALE terminate faster than other SBSE tools?
- **RQ2 (quality):** Does GALE return similar, or better, solutions than other SBSE tools?

Please note that we now list these questions in the introduction and that our new results section is based on these two questions. As to the half dozen research questions in the last draft, there were confused and we have replaced them with the above two.

Overall the paper is well-written. A few grammar-related errors are listed as follows. (omitted for simplicity and space).

We thank you for these corrections. Note however that after a revision, many of these have gone away and otherwise have been corrected.

Reviewer: 2

(Please note that Reviewer2 listed several typos and minor editorial issues that have been addressed in this draft.)

To begin our reply to your comments, we wish to start with one of your latter points:

There are basically two fundamental questions regarding GALE, a first concerning the number of evaluation compared to that of other algorithms and a second concerning the quality of the results it returns (does it find roughly the same solutions as other algorithms?).

You are correct. Your proposed sequence is now the overall guiding principle of this paper- see page2, page8, page9, page10.

Returning now to the start of your comments...

The results claimed by this paper are impressive: if true, it would be a major breakthrough in MOEA with high impacts both in search-based software engineering and multi-objective optimisation in general.

Thank you for that comment.

This result is however hard to validate because the presentation is inadequate. In Section 4, the new MOEA algorithm is not described clearly enough to allow other researches to understand and implement it. The algorithm is actually never shown in full. (Section 4.1, called "Details", give details about the validation experiment rather than details of the algorithm; this information should be moved to the experiment section.)

We agree- this draft now has

1) A high-level summary of the new algorithm, page2;
2) Detailed pseudo-code of the new algorithm on page4, page5, page6.

The POM3 model, which is said to be presented in this paper for the first time, is presented only very summarily. There is not enough information to understand the model, what decisions it supports (Fig. 4 is not enough; it needs explanations), and why its three objectives are valid (I'm surprised by the objective to minimise idle rate which appears to contradict a fundamental principle of lean development which is to maximise value creation instead of resource utilisation).

We agree that there was too much POM3 in the last version- now it is just one of the eight models explored to test the new algorithm.

As to your specific point (that POM3 should be about resource utilization) please note: that is actually in the current version of POM3. Requirements development cost is measured using programmer salaries. We also have an *Idle rate* measure which we need to minimize. So the current goals of POM3 are to make the most use of the programmer's busy time (when working on some current requirement) while minimizing their wasted time (measured as the *Idle Rate* when they are forced to wait on the delivery of other requirements).

It is actually not clear why POM3 is presented in the paper. If the paper's core contribution is GALE, wouldn't it be simpler and better to demonstrate GALE's benefits on previously published models in search-based software engineering? One of the difficulties in reading the paper is that it mixes two novel contributions, GALE and POM3, without clearly distinguishing them, neither in their presentation nor in the experiments. Maybe this work needs to be split in two separate papers, one about GALE and one about POM3, each describing and validating its contribution in full details.

We agree- and this draft was written as per your direction. This paper is now mostly about certifying the GALE algorithm (via a comparative assessment with rival algorithms).

The validation section (Section 6) does not present the results clearly. Given the strong claims in introduction, one would expect a clear and direct comparison of GALE with respect to other MOEA on a set of problems.

We completely agree. The results section of the last draft has been significantly rewritten and simplified. As to the research questions of the last draft, they were poorly formed and so are not used in this draft.

The paper's arguments why it is important to reduce the number of evaluations performed by a MOEA are weak and partly incorrect (Section 2).

We agree- we have now dropped all of that material in this draft.

The claimed improvements of GALE over NSGA-II and other MOEAs is very impressive. Often such improvements are only possible by compromising something but I could not identify whether or what compromises are being made in this case. Are there limitations to the use of GALE over NSGA-II and other MOEAs? Do we lose something by using GALE instead of NSGA-II?

In response to this question, we have tried many tests-

even printing out and manually inspecting all the Pareto frontiers generated by all these MOEAs on all these models (for one sample of that print out, please see the last figure in the paper).

To date, we have not found some compelling reason not to use GALE- which is not to say that the model we use tomorrow might raise novel issues that would mean we have to revise our optimism for this new algorithm (a point that we stress in the "Threats to Validity" section on p11).

Reviewer: 3

(Please note that Reviewer3 listed several typos and minor editorial issues that have been addressed in this draft.)

To begin our reply to your comments, we wish to start with one of your latter points:

What I am really missing (and should I had found it in the paper, I would have recommended a no-doubt accept) is a replication package, with the implementation of the algorithms, and a dataset to test the different algorithms, possibly with suggestions and guidelines about how to apply GALE with other models (for instance, elaborating more on what "engineering judgment" is necessary to adapt GALE to other models).

That replication package is now available on the web (see Section 1.2 on page2). That replication package comes with most of this study's models (sadly, the CDA model is a specialized NASA product that we cannot release to the public).

As to how to apply GALE to different models, please see the start of section 3 (top left of page5) listing the "glue" functions that connect GALE to a model.

Returning now to the start of your comments...

This paper presents a multi-objective optimization algorithm for software projects, called GALE. It is tested with the POM3 model, providing results that could be useful for the management of software agile projects. GALE is compared with other algorithms to assess that it requires less iterations to obtain a result.

The paper presents the following points in favor and against, that we will comment with more detail below:

Pros: (1) New algorithm more suitable for software engineering than other alternatives; (2) Tested with a model that seems to be a "real world" example (3) Example of application of the algorithm included in the paper

Thank you for your kind words. Please note that this draft now has case studies on *two* real world models (POM3 and the NASA CDA model on avionics safety).

Cons: Research questions are more focused in the POM model than in the algorithm

We agree. Those research questions were poorly defined and are not reused here. Instead, we follow the advice of Reviewer2 who wrote "There are basically two fundamental questions regarding GALE, a first concerning the number of evaluation compared to that of other algorithms and a second concerning the quality of the results it returns (does it find roughly the same solutions as other algorithms?". Accordingly, those two fundamental questions are addressed in this paper- see page2, page8, page9, page10.

Not clear whether the decisions made by GALE are better or worse (or none) than those suggested by other algorithms:

This is our new RQ2 (quality). We think Figures 15,16,17 show that GALE's decisions can be better than those found by other algorithms.

As described in section 3.4, GALE has seven tuning parameters that must be set in order to apply the algorithm. However, in section 5, when GALE is assessed, not all these parameters are set. In particular, where is the delta parameter?

We were certainly guilty of over-specification in the last draft. GALE has been significantly simplified for this new draft and now it uses just the three tuning parameters listed in Section 6.3

GALE is already difficult to understand, and this mismatch has left me puzzled wondering how GALE is supposed to be used.

You are correct- our previous description was far too complicated. This draft now has

1) A high-level summary of GALE, page2;
2) Detailed pseudo-code of the new algorithm on page4, page5, page6.

Moreover, these tuning parameters are not explained with enough detail. It is not clear how these parameters must be chosen. The authors just say that they selected using their "engineering judgement", but I think that is very vague to say so. After all, you are comparing against other methods. How do I know that you have not tweaked the parameters to perform better in this particular setting? (with this model, against the other particular algorithms) How do I know that the better performance will hold in other scenarios?

You raise two important questions:

- In section 6.3, we note that we froze GALE's tuning parameters after some exploration of the smallest model (and that we did so before starting our work on NSGA-II and SPEA2).
- As to your other question (will these work for all scenarios), as we say in Section 6.1, there is no general answer to that issue. Hopefully, now that our replication package is on-line, we can work with other researchers to define some context space within which we'd recommend GALE, and outside of which we would not.

Now I would like to comment on the research questions posed in the paper. There are seven research questions and some of them seem to be very focused on POM3 as a model for software projects, rather than on the features of GALE compared against other MOEAs. I can extract one very clear conclusion out of those questions: GALE is faster. I grant you that. But it is

not clear at all how the quality of the decisions made by GALE is compared against the decisions made by the other algorithms.

Those research questions were ill-formed and we do not reuse them here. Instead, we use the research questions based on your comments (see the research questions **RQ1** and **RQ2** defined on page2 and discussed on page8, page9, page10, page11.

As to the comparative quality of GALE's solutions w.r.t. other models, as mentioned above, we believe Figures 15,16,17 show that GALE's decisions can be better than those found by other algorithms. But, what is your view?

Because all of these three points (how to apply GALE? what's the impact on decisions of the algorithm compared to existing options? the lack of replication package and application guidelines), I recommend a major revision.

Please advise: has this draft addressed the matters you raised here?