This is the detailed response to the three reviewers for the paper titled "..." by "...". To help follow all the improvements applied to the paper, we include here the original reports of the reviewers. For each aspect brought up by a reviewer, we explain how we modified the paper to reflect the proposed change. This response is split in three sections, corresponding to each of the three reports, in the order we received them from the editor. It should be noted that the paper was thoroughly revised by the authors to improve the english use and incorporate some of the suggestions given by the reviewers of the masters thesis.

Reviewer 1

The article describes a method of decomposing a graph G into several sub-graphs, in which the maximum clique problems (MCP) are solved and show that they can gain the maximum clique (MC) of the whole graph G by the solutions of the sub-graphs. This result holds whenever exact algorithms for the MCP are used inside the sub-problems. They also provide a new heuristic, so-called penalty-evaporation (P-E) heuristic, for finding large maximal cliques. The combination of the decomposition with the P-E heuristics shows promising results in the experimental section.

I can recommend the paper for publication after several (sometimes major) adjustments are made:

I have the following suggestions:

On p.4-p2-12 the authors write about exponentially large number of cliques. First this is not true in general, second they should write exponentially in _what_ variable and third they should refer to some literature (e.g. moon/moser(65), On cliques in graphs, Isr. J. Math., 3, 23--28)

We modified the paragraph to be a bit more precise and added the reference. It should be noted that a complete graph is a clique of size |V|, but the total number of cliques in this graph is the number of all the possible combinations of vertices, which is exponentially large in terms of |V|. An enumeration procedure has to, in some way, enumerate and discard all these cliques in order to find the largest one. Of course, they include some mechanism to avoid unnecessary computations, but they still enumerate (and discard) a large proportion of non-optimal cliques.

It should be mentioned that solving the maximum independent set problem is equal to solving the MCP in the complementary graph (the terms are mixed up in the lit. Rev.)

We modified the literature review and added a precise definition in the first paragraph of the literature review.

On p.6,p3: they describe genetic approaches to the MCP in this context the work of I.M.Bomze and V. Stix(99), Genetic engineering via negative fitness: Evolutionary dynamics for global optimization, Annals of Oper. Res., 89, 297--318 would fit in, because it combines the genetic approach mentioned in this paragraph with a strategy (mentioned earlier on p.4) in order to escape local inefficient solutions, which delivers an exact method for the MCP.

We added the reference in the proper section of the literature review.

On p.6, sec 3: The decomposition section comes out of the blue without motivation after the literature review. Maybe on or two sentences could help (but see also below regarding the structure of the paper)

We added a small paragraph at the end of the literature review, just before the decomposition section, where we introduce a reference to Babel's branch and bound algorithm and we point out that our decomposition method is in the same spirit.

On p.6,p2,1 1-3: "at each iteration…". It is not clear that the large clique is especially constructed, such _that_ its vertices do not belong to any other larger one.

This becomes clear in the algorithm. The purpose of this sentence is to introduce an overview of what is to be developed in the algorithm.

Algorithms: the algorithms are not very clear. At least there should be a line saying what is the result of the algorithm and in which variable can it be found.

We added some more explanations in the paragraph preceding the algorithm.

On p.6,theorem: "(i.e. the algo..." should be like "(i.e. the algorithm finds one maximum clique inside G'(i)"
cont.: "then the decomposition ... generates an optimal solution" What is the optimal solution (again maximum clique should be used) and where can this solution be found? What is the output of the algo? (see above)

We reformulated the preceding and following paragraphs to include these specifications.

on p.6,proof: at first sight it seemed to me that the proof consists of two lines only. The paragraphs should be held together or a end-of-proof symbol would be nice.

We reformulated the proof in a single paragraph, and added a **Q.E.D** at the end of the paragraph.

The algorithm itself reminds me of the algorithm of Babel (reference 2). Babel also iterates through vertices and looks inside the neighborhood of them for maximum cliques. There are some other differences but maybe the author should point them out.

At the end of Section 3 (the decomposition section), we compared both approaches and mentioned the similarities and differences.

Secondly what I am really missing is a recursive approach of the decomposition algorithm (as Babel's decomposition offers). Step 2.1 of the algorithm (find the MC inside G'(i)) is practically impossible when using exact algorithms, because the decomposition method does only a single decomposition on the first level. In dense graphs the decomposed graphs G'(i) will have almost the same order than the original one. Therefore I ask the authors if they have thought about a recursive application of the decomposition in order to apply some exact methods once the graph is really smaller. Maybe that's the reason why only heuristics are applied to the decomposition in the experimental section.

In the paper we mention that it would probably be better to embed an exact method to solve large *sparse* graphs, and embed a heuristic method to solve large *dense* graphs, but further analysis for the exact case clearly surpasses the scope of this paper.

On p.7, algorithm: maybe only step 3 should be written down, because its the only step that differs from the prev. Algo.

We reformulated the algorithm and added some explanations in the surrounding paragraphs.

The P-E approach (section 4) is well done and there are a lot of interesting ideas in it. Again on p11 the algorithm could be a bit more self-explanatory (what is the in/output)

We added some specifications in the preceding paragraph.

all tables in the article should be more self-explanatory. E.g. I would suggest to write the numbers in table 5.1 5.2 ... in a percentage of success or failure instead of absolute numbers. The caption of the tables and the labels (especially tab 5.5 5.6 5.7) should be improved. It is very hard to find the information within the mixture of text and tables.

We changed the values into percentages, adapted the surrounding text, and improved the captions of the tables.

Final comments:

Because the main results and experiments concentrate on the P-E algorithm I would suggest to start with that one and incorporate the decomposition into the P-E algorithm afterwards.

We considered this option, but discarded it because the P-E algorithm is embedded in the decomposition scheme, and introducing it before the decomposition provokes a change in focus that tends to lose the reader. The main advantage of introducing the decomposition algorithm first is to allow the reader to start with an overview of the overall algorithm and then progressively focus on the details. Also, when the reader reaches Section 5 (results), he is already familiar with the penalty and evaporation process (which he just read) and it is easier to follow the calibration phase.

I do not think that the presented decomposition works well on exact methods because of the reasons mentioned above.

As mentioned in an above response, we did mention in the paper that embedding an exact method in the decomposition algorithm would be a good strategy to solve large sparse graphs, but embedding a heuristic method would be better to solve large dense graphs.

I wander how the P-E algo. would succeed using other easy decompositions like e.g. the one suggested by Babel (on first level only).

It would be interesting to know indeed, but we think this is beyond the scope of the paper.

It should be distinguished between finding the MC and proofing that the found solution is the MC, which is much harder. All algorithms described in this article are only able to find the MC. If the size of the MC, however, is unknown the deliver only a lower bound on it.

We added the precision in the fourth paragraph of the introduction.

Because the main results and experiments concentrate on the P-E algorithm I would suggest to start with that one and incorporate the decomposition into the P-E algorithm afterwards.

We considered this option, but discarded it because the P-E algorithm is embedded in the decomposition scheme, and introducing it before the decomposition provokes a change in focus that tends to lose the reader. The main advantage of introducing the decomposition algorithm first is to allow the reader to start with an overview of the overall algorithm and then progressively focus on the details. Also, when the reader reaches Section 5 (results), he is already familiar with the penalty and evaporation process (which he just read) and it is easier to follow the calibration phase.

I do not think that the presented decomposition works well on exact methods because of the reasons mentioned above.

As mentioned in an above response, we did mention in the paper that embedding an exact method in the decomposition algorithm would be a good strategy to solve large sparse graphs, but embedding a heuristic method would be better to solve large dense graphs.

I wander how the P-E algo. would succeed using other easy decompositions like e.g. the one suggested by Babel (on first level only).

It would be interesting to know indeed, but we think this is beyond the scope of the paper.

It should be distinguished between finding the MC and proofing that the found solution is the MC, which is much harder. All algorithms described in this article are only able to find the MC. If the size of the MC, however, is unknown the deliver only a lower bound on it.

We added the precision in the fourth paragraph of the introduction.

Reviewer 2

The paper is very well written and gives a short and precise introduction to the topic. Their contribution is very well explained and they get impressive results. Their decomposition method clearly shows its benefits to improve the underlying heuristic. Their local search heuristic P-E alone performs almost as good as tabu search. **Since tabu search often gets better results than P-E alone, it would have been interesting to know the performance of a decomp-tabu algorithm. Unfortunately this has not been discussed in the article.**

It would be interesting to know indeed, but we think this is beyond the scope of the paper. We can still approximate the results by comparing P-E and MIN when embedded in the decomposition method. Indeed, P-E is a bit slower than MIN (since it does a bit more iterations) but gives better results. When embedded in the decomposition method, both the time and the quality of the results are multiplied by some factor. A similar behavior is foreseeable when embedding a tabu search in the decomposition algorithm. Times would be larger than P-E embedded in the decomposition method, but the quality of the results should also be improved.

The two proposed procedures are general enough to be adapted to other problems. Due to this fact and the results obtained the article is of great interest to the research field.

I recommend this article to be published without modification.

Reviewer 3

The authors present a new heuristic approach to the solution of the maximum clique problem along with a decomposition scheme to reduce the size of the processed problems. The basic algorithm can be viewed as a modified tabu search; however, rather than prohibiting the selection of a specific move, such selection is made more difficult by adding a penalty term to its fitness value. Large penalties are equivalent to a full prohibition. The penalty associated to a move is reduced at each iteration (evaporation). This "penalty-evaporation" heuristic is applied to the solution of the subproblems obtained by applying the decomposition scheme to the original instance. The decomposition is based on the following, simple principle: if C is (the vertex set of) a clique, and no larger cliques are contained in the subgraph induced by C and its neighborhood, then C can be removed, and the search can be carried on the residual graph. This scheme is exact, but can be relaxed (with minor amendments) by contenting ourselves with finding large cliques. The algorithm has been tested over the DIMACS instances, and the computational results appear to be competitive with the best algorithms in the literature.

In my opinion the paper can be accepted after revision. Even though the ideas developed are not too deep, still the algorithm favourably compare with the best algorithms presented in the literature; also, the experiments have been carefully carried out on the DIMACS test-bed, which is good. The bibliography section is sufficiently rich. Both the english usage and the mathematical formalism should be improved in the revised version. A main observation concerns the decomposition mechanism, which plays a crucial role in enhancing the algorithm. **However, it is unclear to me how this decomposition has an effect on the difficult instances (the dense ones). Specifically, while executing Step 2 for the first time the algorithm will probably explore a large portion of the graph. In other words, if the graph is dense, it is very likely that a largest clique contains one of the vertices in the first clique(s). I think some computational results on a pair of dense instances may be of help. In particular, the authors should report the size of the cliques found at each iteration and the size of the neighborhood of such cliques. Also, a set of experiments (even on randomly generated graphs) to relate the efficacy of the decomposition versus the density of the instances should be included.**

We gather from these comments that we need to show the usefulness of the decomposition method, particularly when the density of the graphs increases. Indeed, the way the decomposition method is working leaves no doubt concerning its efficiency on sparse graphs. But, after reading these comments, we also felt the need to show numerically the efficiency for dense graphs. To do so, we randomly generated a large set of graphs (300 to be precise) of various sizes between 100 and 500 vertices and of densities 0.5, 0.7 and 0.9. Then we compared the results obtained by executing the P-E method alone versus the P-E method embedded in the decomposition method. We noted the number of times embedding the P-E method in the decomposition algorithm improves the size of the largest clique found. The results show that the decomposition method gains in effectiveness as the graphs grow in size and as the density increases. .

We added these results and the comments at the end of the results section.

It follows a list of specific comments.

1. **Page 1, Line -14: The decomposition scheme is not a solution approach. More correctly the authors should introduce one solution approach along with an effective decomposition scheme.**

We applied the modification as proposed.

2. **Page 1, Line -6: I do not think that the definition *greedy-like* fits with the presented algorithm. In fact, both the formal definition (based on matroid theory) as well as the practical and standard usage of the term involve some type of irrevocable choice (i.e. an element is included in the solution and never removed). On the other hand, most of known heuristics include some type of greedy choice (i.e. an element maximizing some weighing function is temporarily included in the solution). See for example [?].**

We modified the presentation of the algorithm to specify accurately the behavior of the algorithm.

3. **Page 2, Line 6: this looks more like an example of application of graph coloring; many other (more fitting) applications are at hand.**

The paragraph was rewritten in accordance to this comment.

4. **Page 2, Line 16: "... There exists ..." change to "... There exist ..."**

We applied the modification as proposed.

5. **Page 3, Line 10: "... ajusting ... " change to "... adjusting ..."**

We applied the modification as proposed.

6. **Page 4, Line -17: the expression "different parts of the graph" (also used other times in the sequel), in this context, is too sloppy and misleading. Still being informal, but a bit more precise, one could use the expression "solution space" instead of "graph".**

We applied the modification as proposed by replacing "graph" by either "solution space" or "feasible domain", according to the context.

7. **Page 5, Line 1-4: Actually, this is not a way to *avoid* the use of a tabu list: it is only a smart way to *represent* a tabu list - as observed by the authors in [21] (in the bibliography section of the paper). There is a confusion here between an abstract data structure and its implementation.**

This part of the literature review was mostly rewritten in order to avoid any confusion.

8. **Page 6, Line 15: the definition of neighborhood (typically denoted by N(i)) is non standard. In general, vertex i does not belong to its neighborhood. Also, the neighborhood is not a graph, but a set of vertices.**

We modified the definitions accordingly.

9. **Page 6, Line 17: "... are adjacent ..." change to "... is adjacent ...".**

We applied the modification as proposed.

10. **Page 6, Line -6: the proof of the main theorem is a bit involved. I propose something like "Let \bar{k} be the smallest k such that C^* intersect C^k \not = \emptyset: then $|C^{\bar{k}}| = |C^*|$. In fact, (denoting by G^k the residual graph at iteration k), C^* is completely contained in $G^{\bar{k}}$."**

The theorem and proof have been rewritten, mostly as proposed.

11. **Page 8, Line 18: see note Page 1, Line -7.**

See response for point 2.

12. **Page 9, Line 8: "... proportionnal ..." change to "... proportional ..."**

We applied the modification as proposed.

13. **Page 12, Line 2: what is an "interesting part of the graph"? Be more precise (see also note Page 4, Line -17).**

We modified the text to add some precisions on the behavior of the decomposition algorithm. We also applied the modification as proposed (see note Page 4, Line -17).

14. Page 12, Line 21: here and later, "best" and "optimal" stands also for "best known". Actually, this is later specified but I think the authors should mention it earlier.

We applied the modification as proposed.

15. Page 15, Line -16: "... independant ..." change to "... independent ..."

We applied the modification as proposed.

16. Page 20, Line 6: "... small values ..." change to "... small absolute values ..."

We applied the modification as proposed.