**LISTED RESPONSE TO COMMENT/SUGGESTIONS OF**

***REVIEWER 2***

|  |  |  |  |
| --- | --- | --- | --- |
| ***No*** | ***Reviewer’s Comment/Suggestion*** | ***Author’s Comment*** | ***Action Taken\*\*\**** |
| 1 | In test results shown in Tables 1,2,3, most results of MOD\*-Lite and MOA\* are the same. The authors should discuss why these two complete algorithms yield different results in some simulations. | Thanks for your comments and suggestions. Results on Table 1 and 2 belong to fully observable environments that are generated either randomly or manually, respectively. As MOA\* and MOD\* Lite are both complete algorithms, it is expected that they will both find optimal or sub-optimal results. That's why almost all of the results are the same. For different results, we can say that the algorithm which find dominatable result finds the sub-optimal one instead of optimal. However, for results on Table 3, which are the ones for partially observable environments, the agents are probable to discover different parts of the environment for MOA\* and MOD\* Lite. This naturally causes different results to occur. | Modifications and additions are done in accordance with author reply in Section V-A and Section V-B. |
| 2 | The column of MOGPP in tables should be described more clearly. | Extra explanation for MOGPP columns in tables is added : “ Also it is important to emphasize that MOGPP does not guarantee Pareto optimum and that's why the results are not optimal or sub-optimal.” | Modifications and additions are done in accordance with author reply in Section V-A. |
| 3 | MOGPP implementation does not show the Pareto front , which is an essential concept in multiobjective problems. | We implement MOGPP as a soft computing alternative genetic realization of multi-objective path planning problem, and we don't guarantee that it will yield optimal results, as this is difficult in solving many optimization problems using evolutionary heuristics. We only state that MOGPP is a complete algorithm, and eventually finds a solution in the search space. That’s why MOGPP results do not show Pareto front. | First paragraph of Section IV is extended to clarification. |
| 4 | The comparison should be made also with single-objective path search algorithms using weighted sum of multiple objective, and it would be more convincing if the MOGPP implementation is from the literature such as SPEA2 or NSGA2. | In this study, we concentrated on multi-objectivity on path planning where objectives that cannot be transformed into each other. Thus, calculating weighted sum of multiple objectives (in case where we transform one objective to another) could be misleading. For example, converting distance metric to danger in our case study is not possible and any weighted summation is debateable.  We implemented SPEA2 algorithm and run all experiments. We realized that SPEA2 is fast for fully observable environments but its path qualities are unacceptably worse. For partially observable environments, SPEA2 realization suffers from exponential time cost similar to MOA\* and MOGPP algorithms. | SPEA2 test results are included in *Section V Experimental Results* and its performance is evaluated. We also noted that SPEA2 test was included in *Abstract, Section I (Introduction) and Section VI (Conclusion & Future Work)*. |