# Comment 1:

The manuscript introduces a software package for optimization written in C++. It is a good contribution as a software tool. It reads like a software manual. It is not clear what the scientific contributions are and what new knowledge we can gain from this manuscript.

# Response

# Comment 2:

There are well-known C++ packages that provide similar features to the proposed one:  
- Paradiseo: https://nojhan.github.io/paradiseo/  
- Pagmo: https://esa.github.io/pagmo2/  
- HeuristicLab (in C#): https://dev.heuristiclab.com/trac.fcgi/  
  
None of these works are mentioned the paper; instead, references such as [60] and [61) are provided, but those papers were published in 1996 and 2004.   
  
The authors must analyze the state-of-the-art and clearly state which are the contributions of the proposed package that are not covered by existing ones.

# Response

# Comment 3:

The authors have included references of other similar tools (Paradiseo, Pagmo and HeuristicLab). However, merely including them is not enough. The proposal should be compared against them and show what aspect not covered by the existing tools are offered by OPTIMUS. An idea would be to select a number of features and include table indicating which ones are covered by each framework; this report could give some ideas: https://www.uco.es/grupos/kdis/mofs-multiobjective/KDIS\_TechnicalReport\_comparison\_of\_metaheuristic\_frameworks\_for\_moo\_2020.pdf. See for example Tables  1 (features), 3 (algorithms) 12 (benchmarks). The authors can take a look also to Paradiseo's Web Page (https://nojhan.github.io/paradiseo/), which contains a section "VS other Frameworks" including two comparative tables.  
  
Important features are whether the tool is easy to use and is well-documented. However, taking a look to the software repository, the full documentation manual only contains instructions to download, install and compile the package. There are not any examples, the content is not clearly structured (it is a plain directory with many subdirectories) and basic information such as the license is missing.  
  
A key issue that is not addressed is related to the extensibility of the framework. How can the users include their own algorithms in OPTIMUS?. Does it offer an object-oriented architecture for algorithms fostering code reuse?   
  
The same applies for the problems: how can they defined? Is there a class hierarchy for continuous, integer and combinatorial problems?. The paper includes an experimentation section with examples, but it contains the problem formulations and tables with results, while what would be interesting is to detail how these problems are implemented.   
  
As for experiments, when comparing metaheuristics it is mandatory to apply statistical tests to assess whether the differences between the compared algorithms are significant. These tests are missing in the paper and a framework for metaheuristics should offer support for calculating them.