# 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

We have the following statement in the introduction:

“*In this paper, a new integrated computing environment for performing global optimization methods for multidimensional functions is presented and analyzed in detail. In this computing environment, the programmer can code the problem to be solved using a high-level programming language such as C++. In addition to the objective function, the programmer can also provide information that the objective problem should have at the start of the optimization process and, in addition, can formulate a series of actions that will take place after the optimization process is finished. Subsequently, the researcher can formulate a strategy to solve the problem. In this strategy, the researcher can choose from a series of sampling methods, choose a global minimization method established in the relevant literature and possibly some local minimization method to improve the produced result.*”

# 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

We have the following statement in the Introduction section:

“*Similar software environments can be found, such as the BARON software package [*[*baron*](#LyXCite-baron)*] for non-convex optimization problems, the MERLIN optimization software [*[*merlin*](#LyXCite-merlin)*] which is accompanied by the Merlin Control Language compiler to guide the optimization course, the DEoptim software [*[*deoptim*](#LyXCite-deoptim)*] which is an R package implementing the differential evolution algorithm, the PDoublePop optimization software [*[*pdoublepop*](#LyXCite-pdoublepop)*] that implements a parallel genetic algorithm for global optimization etc.*

*Also recently, some other optimization tools have appeared such as the Paradiseo [*[*paradiseo*](#LyXCite-paradiseo)*] implemented in C++, which mainly includes evolutionary algorithms, the Pagmo software [*[*pagmo*](#LyXCite-pagmo)*] where a wide range of evolution algorithms are incorporated to solve optimization problems, and finally another approach for evolutionary algorithms applied to optimization problems is the HeuristicLab freely available from https://dev.heuristiclab.com/trac.fcgi/, used mainly for online optimization. In the proposed software, the user can write the required objective function in simple C++ and then choose from a wide range of global optimization methods, the most suitable one for finding the global minimum. Furthermore, in the proposed software, the user can parameterize the local minimization method to be used as well as the termination method to be used for the successful termination of the technique. In addition, it is possible for the user to create his own global minimization method from scratch using the programming tools of the Optimus libraries.*”

# 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.

# Response

A)We have included the license file in the distribution, located in <https://github.com/itsoulos/GlobalOptimus>/license.txt

B)We have included instructions in the manuscript as well in the distribution <https://github.com/itsoulos/GlobalOptimus/README.md>for

1) Linux based systems

2) Bsd based systems

3) Windows based systems

C)The content of the distribution has been changed, with subdirectories indicating the content, such as METHODS, PROBLEMS etc.

D)We have added the subsection 2.4 entitled “Implementing a user - defined optimization method” with instructions to add a user – defined optimization method. Also, in the same subsection we have added a flowchart that indicates the execution flow of any optimization method.

E)We have added the subsection 2.5.2 entitled “User defined problem”, where we explain how the user can code his own objective problem.

F)We have the following experiments in the revised manuscript

1)Experiments with some test functions of the relevant literature. In this experiment we have added a statistical comparison using boxplots.

2)Experiments with the Lennard Jones potential and various optimization methods. We have also added a graph in this experiement to outline the superiority of the iPso method.

3) Experiments with Parallel Optimization

4) Experiments with the 0.1. Fuchs-Kliewer polaritons and the incorporation of Genetic Algorithms.

5) Experiments with the Sampling Option.