OpenSolver LibreOffice Port

Asher Gingerich, Donghun Ha
Department of Computer Science & Engineering • Taylor University • 236 W. Reade Ave, Upland, IN 46989

Introduction

OpenSolver is an extension for Microsoft Excel that operates as a linear programming solver with many different settings and user experience improvements. This extension is used extensively in Dr. Geisler's Operations Research. In the 2018 Fall Semester, this extension stopped working on macOS environments due to an update to Excel.

This project is the solution to that problem. Porting OpenSolver's functionality to LibreOffice should allow a unified way to access the solver, since it is free and open source and thus easily accessible to students. OpenSolver is originally written in Visual Basic for Applications (VBA), and is only partially supported within LibreOffice –which primarily uses LibreOffice Basic.

Macro Translation

The first step in porting the code was understanding how the code was all connected. Functions that depended on the existence of other functions or values were diagrammed to provide a basic order of implementation, from the most independent to the least independent.

After diagramming, the actual porting of the code began. This was more difficult than it appeared at first glance due to incompatibilities between the two languages. The partial support for VBA within LibreOffice allowed many functions to almost be directly ported, but there were some important features that were missing from this support that inhibited development, and slowed the process down.

User Interface

Both Visual Basic and LibreOffice Basic provides forms and dialogs – GUI(Graphical User Interface) tools that are intuitive to use for both developer and users. With forms and dialogs, developers can create a pop-up screen that could take various types of inputs such as plain texts and enumerated data.

Our team created a demo solver UI that would resemble existing solver UI so that it would be easy to use for anyone who used OpenSolver before. Continuous improvement of UI was processed throughout the project term to implement more functionality for better user experience.

	A	B	C	D
1		Target Cell		
2	Maximize:	20000		
3				
4	By Changing Cells:			
5	Var1	20		
6	Var2	35		
7	Var3	100		
8				
9	Const 1	4700	<=	5000
10	Const 2	300	>=	300
11	Const 3	6000	>=	6000

Figure 1. Demo solver sheet

Future Work

In the future, groups will need to fully implement the various solver types that exist within the original OpenSolver extension, which will also involve implementing IO functionality. Sensitivity Analysis will also need to be implemented, along with improving UI interaction, so that it would closer resemble OpenSolver, with features such as selecting cells with the mouse during dialog input.

Conclusion

Working on this project required skilled experiences in Visual Basic and LibreOffice, which were the languages we were not familiar with before this project. Throughout this project, our team learned not only the languages and appropriate programming skills, but also how to deal with tasks that requires new technology. In general, this has been a great learning process.

Despite the limitations and challenges we encountered, our team made a meaningful progress in the porting process. We successfully ported significant amount of the codebase from OpenSolver, and created a functional demo user interface that has some of the important features our client requested.

While there is some major improvements need to be made before use, we believe our porting process has provided solid foundation to this project, and we hope that in the future it is helpful to the scholars and students concerned with Operations Research.