

A Unified Optimization Framework for Microelectronics Industry

Yiming Li

Department of Communication Engineering, National Chiao Tung University, 1001 Ta Hsueh Rd., Hsinchu 300, Taiwan

yml@faculty.nctu.edu.tw

Cheng-Kai Chen

Department of Communication Engineering, National Chiao Tung University, 1001 Ta Hsueh Rd., Hsinchu 300, Taiwan

ckchen@mail.ymlab.org

Yen-Yu Cho

Department of Communication Engineering, National Chiao Tung University, 1001 Ta Hsueh Rd., Hsinchu 300, Taiwan

yycho@mail.ymlab.org

ABSTRACT

In this paper, an object-oriented unified optimization framework (UOF) for general problem optimization is proposed. Based on evolutionary algorithms, numerical deterministic methods, and C++ objective design, the UOF possesses real-world applications for various optimization problems. The UOF is with interface between defining a general problem and generic solver; hence the two areas are bridged. The UOF's components are categorized into problem and solver parts, and they work independently. Therefore, the reusable high level code allows the adaptation to new problem and solver quickly. Practical applications in microelectronics industry confirm the functionality of the developed unified optimization framework.

Categories and Subject Descriptors

[Real-World Applications]

General Terms

Algorithms, Performance, Design, Languages, Verification

Keywords: Evolutionary and deterministic methods, Optimization, Traveling salesman problem, Reverse modeling problem, antenna, simulation-based optimization.

1. INTRODUCTION

For a general purpose optimizer, defining a comprehensive interface between problems and solvers is one of important issues in programming language [1-2]. In public domain, different optimization frameworks, such as GALIB, DESMO, and NP-Opt. have shown their advantages. However, it still has room to improve the interface between problem and solver for real-world applications. In this paper, we implement a C++ unified optimization framework (UOF) for general problems and solvers. Three distinct problems, (1) the traveling salesman problem, (2) the semiconductor reverse modeling problem, and (3) the optimal antenna shape problem for wireless communication, are demonstrated to show the performance and extensibility of the UOF. The preliminary results confirm that the UOF solves these problems in a cost-effective way.

2. A UNIFIED OPTIMIZATION FRAMEWORK

All members in UOF are categorized into problem related and solver related. In the problem relative category, classes are used to form a problem in the UOF, and so are those in the solver relative category. The UOFProblem class is the main class to define the problem; as for UOFInitializer class, it is responsible for the initialization of the solution for each problem; The UOFEvaluator class provides the method of evaluating the result obtained by UOFProblem derived classes; The UOFConstraint class defines the type and candidate or range of each parameter. On the other hand, the UOFSolver class takes the major character in solver relative category, and the UOFSolution class stores possible solutions and several operators of the specified solver; class UOFTerminator judges when and how to stop the optimization process of the solver; and UOFInfo class logs the behavior of the solver during the solving process. All classes in both categories are abstract classes, and need to be implemented in derived classes.

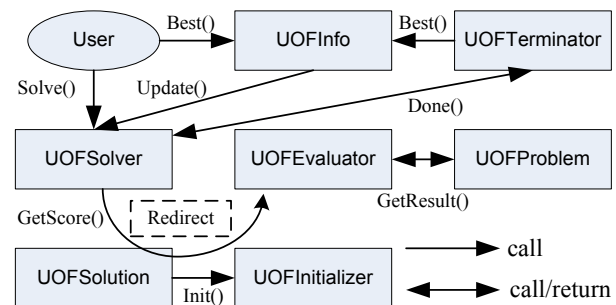


Figure 1. The working flow and object inner connections in the UOF from users' point of view.

Figure 1 shows the working flow and object inner connections in the UOF from users' point of view. At the beginning, users call *Solve* function in UOFSolver class to start the optimization operation. Meanwhile, the UOFInitializer initializes the UOFSolution objects. The *Solve* function invokes *GetScore* function in UOFSolution class to get the results with given parameters. The UOFSolution class redirects and passes the *GetScore* message to UOFEvaluator class. The subclass of UOFEvaluator object is a function object which behaves like a function pointer in C language but takes advantages of C++ object-oriented programming. The UOFEvaluator object passes the solution to UOFProblem, and calls *GetResult* to get the

corresponding results. While the solver solves the problem, in each iteration the UOFInfo updates the current status of the solver, such as the best parameter set, moreover, the UOFTerminator also update the current best solution from UOFInfo, and tells UOFSolver whenever to stop the optimization procedure. Finally, users can obtain the best solution from UOFInfo object.

Table 1. The optimized result of the TSP.

Method	geometry	# of cities	Average Iteration	Percent of success run
GA	Circle	50	6562	30%
ACO	Circle	50	18	60%
GA	Matrix	25	3600	30%
ACO	Matrix	25	26	100%

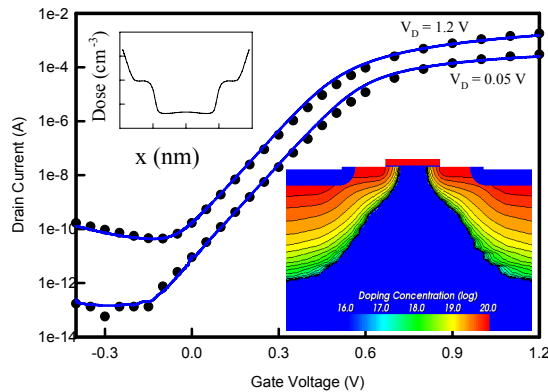


Figure 2. The achieved accuracy of the extracted I-V curves for 65 nm N-MOSFET. Results are simultaneously obtained with considering device and process configurations. Symbols are measured data and lines are optimized result.

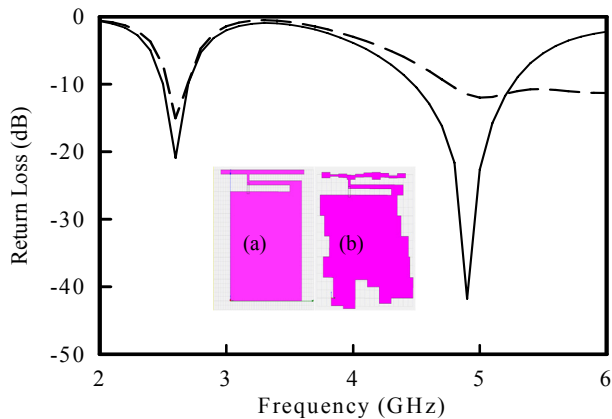


Figure 3. The return loss of the antenna. The inset plot is the (a) original and (b) optimized shape of antenna, respectively.

3. APPLICATIONS AND RESULTS

In this section we demonstrate applicability of the UOF by solving the traveling salesman problem, and two simulation-based optimization problems in microelectronics industry.

3.1 The Traveling Salesman Problem

In this experiment, two geometry distributions of cities are examined. The first distribution forms a circle with 50 cities, and the second distribution is a 5x5 matrix. The genetic algorithm (GA) and

ant colony algorithm (ACO) are performed to compare the efficiency for this problem; results are shown in Tab. 1. As shown in Tab. 1, the ACO shows more capability than GA on this problem, which confirms the performance of the UOF for discrete combinational optimization problem.

3.2 The Device Reverse Modeling Problem

For a set of given current-voltage (I-V) curves, shown in Fig. 2, the semiconductor device reverse modeling problem features finding an optimal doping profile, shown in the inset of Fig. 2, to meet the design specification. Semiconductor process and device simulations [3] are incorporated with UOF. The optimized doping profile and a cutting-plot along channel are shown in the inset of Fig. 2. This examination confirms the robustness of the UOF.

3.3 The Optimal Antenna Design Problem

The original shape of the examined antenna is shown in Fig. 3a, and the return loss plotted with dash line does not meet the desired specification. A final optimized shape is shown in Fig. 3b and the optimized return loss (solid line) meets the goal. The working frequency is at 2.6 and 5 GHz. They are for mobile broadcasting and the 802.11a WLAN protocol.

4. CONCLUSIONS

In this paper, an object-oriented UOF for general problem optimization has been presented. We have successfully customized the UOF to deal with three problems: the TSP, the semiconductor reverse modeling problem, and the design of optimal shape of antenna. Results have confirmed the functionality of the proposed UOF. More problem categories and solvers will be implemented and tested in future work. Advanced optimization methods, such as hybrid computational methods are currently under investigation.

5. ACKNOWLEDGMENTS

This work was supported in part by National Science Council of Taiwan (NSC) under Contract NSC-94-2215-E-009-084 and Contract NSC-95-2752-E-009-003-PAE.

6. REFERENCES

- [1] Wall, M., (2000) GALib a C++ Library of Genetic Algorithm Components. Mass. Inst. of Technol., Cambridge. [Online]. Available: <http://lancet.mit.edu/ga>.
- [2] Gehlsen, B., Page, B., A framework for distributed simulation optimization, *Proceedings of the 2001 Winter Simulation Conference*, pp. 508-514.
- [3] Li, Y., and Sze, S. M., Chao, T.-S., A Practical Implementation of Parallel Dynamic Load Balancing for Adaptive Computing in VLSI Device Simulation, *Engineering with Computers*, 18, (2002) 124.
- [4] Yu, S.-M. and Li, Y., A Pattern-Based Domain Partition Approach to Parallel Optical Proximity Correction in VLSI Designs, *Proceedings of 19th IEEE International Parallel and Distributed Processing Symposium*, 2005.