Team Amalgam

# SE390 Research Plan

Joseph Hong, Chris Kleynhans, Ming-Ho Yee, Atulan Zaman
{yshong,cpkleynh,m5yee,a3zaman}@uwaterloo.ca

November 28, 2012

**Abstract**

SE390 Research Plan for Team Amalgam

## Contents

**1 Problem Definition** **2**

**2 Related Work** **3**

**3 Research Value** **3**
3.1 Aerospace . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
3.2 Civil Engineering . . . . . . . . . . . . . . . . . . . . . . . . . 4
3.3 Software Engineering . . . . . . . . . . . . . . . . . . . . . . . 4

**4 Goals** **4**

**5 Methodologies** **4**

**6 Risks & Technical Feasibility** **5**

**7 Costs** **6**

**8 Legal/Social/Ethical Issues** **6**

**References** **6**

# 1 Problem Definition

Multi-objective optimization is a widely researched area of computer science that focuses on finding solutions to problem definitions with respect to given objective realization constraints. Computing such solutions is extremely resource intensive, and the computation time grows exponentially with the number of optimization variables.

The nature of our work in scientific terms is called *exact, discrete multi-objective optimization*. Multi-objective optimization (MOO) is the process of computing the most optimal solutions given a goal and a set of constraints.

*Multi-objective* means that there are multiple metrics that must be optimized over. Thus, more than one optimal solution may be found that satisfies the various optimization goals.

*Exact* in the context of our problem indicates that all solutions computed by the algorithm are Pareto-optimal. Informally, this means none of the solutions can be improved without compromising at least one other optimization goal. Heuristic approaches and genetic algorithms for multi-objective optimization do not guarantee that all their solutions are Pareto-optimal.

*Discrete* indicates that our optimization algorithm only addresses discrete, or combinatorial, input data. The algorithm does not accept continuous optimality conditions.

One simple example of a multi-objective optimization problem is the satellite scheduling problem. In this problem [1], NASA must determine the best possible satellite launch schedule that maximizes scientific value. Each satellite has a different cost, purpose, and value to a different scientific community. In this problem, the constraints for NASA are such things as resource limitations and launch ordering constraints.

Moolloy [2] is a tool introduced by the MIT Computer Science and Artificial Intelligence Laboratories that solves multi-objective optimization problems. It also has a GUI that lets the user specify the constraints and objective conditions, as well graphically view the Pareto-optimal solutions computed by the algorithm. The underlying algorithm, called the *guided improvement algorithm*, is capable of solving small problems; however, its scalability is greatly limited by the time it takes to compute solutions for large problems.

Therefore, our work will focus on increasing the performance of Moolloy, while ensuring the algorithm continues to perform correctly.

## 2 Related Work

The *guided improvement algorithm* was described by Rayside, Estler, and Jackson [2] in 2009. In their paper, they also conducted a literature survey on related work. Rayside et al. found that most multi-objective problems were concerned with continuous variables, as opposed to discrete variables.

Furthermore, most of the research on multi-objective optimization they found was focused on heuristic approaches, specific instead of general solvers, extensions of single-objective solvers, or problems with only two or three variables.

The guided improvement algorithm is an exact, discrete, general-purpose solver. Furthermore, it is not an extension of a single-objective approach. Rayside et al. identified a similar approach they call the *opportunistic improvement algorithm*, which was independently discovered by Gavanelli [3] and Lukasiewycz, Glaß, Haubelt, and Teich [4]. Notably, the guided improvement algorithm produces intermediate Pareto-optimal results during its computation.

Only three publications have cited the guided improvement algorithm paper since it was published. One of them describes a spreadsheet-like user interface, while the other two concern applications of the algorithm. Thus, none of them are related to our work, which is strictly to optimize the algorithm.

A more recent paper by Dhaenens, Lemesre, and Talbi [5] proposes *K-PPM*, an algorithm which can be parallelized. However, the algorithm does not produce intermediate Pareto-optimal results.

## 3 Research Value

Multi-objective optimization problems appear in may different fields. By improving the performance and scalability of the algorithm, we will enable its usage for problems with larger input spaces. We have identified potential case studies from three different fields: aerospace, civil engineering, and software engineering.

### 3.1 Aerospace

Every ten years NASA performs its decadal survey to determine the satellite launch schedule for the next decade [1]. Multi-objective optimization can be used to determine a schedule that maximizes scientific value to different scientific communities while minimizing cost. Furthermore, many different constraints must be satisfied. For example, one satellite may depend on another, or a satellite may need to match a specific timeline.

### 3.2 Civil Engineering

Dr. Bryan Tolson, a civil engineering professor from the University of Waterloo, has identified a number of multi-objective optimization problems in his research. Currently, the problems are solved using heuristic methods or genetic algorithms. As discussed earlier, these methods do not guarantee that their solutions are Pareto-optimal. One of Dr. Tolson's problems is to determine the optimal type, quantity, and arrangement of materials to line a landfill in order to minimize seepage, while keeping cost minimal.

### 3.3 Software Engineering

We will be collaborating with Rafael Olaechea, a graduate student at the University of Waterloo who is interested in the *software product lines* problem [6]. This problem involves determining which modules should be included in the software for an embedded device. Each module can perform different functions, which may conflict with other modules. Additionally, each module will have a different cost in terms of code size and performance metrics. Therefore, the problem is to determine an optimal set of modules for a device that requires certain functions.

## 4 Goals

The goal of this project is to reduce the computation time for Moolloy of large problem spaces with many optimization goals and to increase the scalability of Moolloy so that by the end of this project Moolloy can successfully computes solutions to optimality problems within a comfortable time bound, that are out of reach in the current version. Part of meeting this goal would mean creating a regression suite to make sure the results we are getting from optimization are still reliable and that might a comparison metric for the new results with the original results during our regular build system.

## 5 Methodologies

The methodologies that we are considering as a logical starting point are the following, which are results of previous works by researchers in this field and also by Derek Rayside and his collaborators. As we proceed with the project it possible we might find some of these optimization ideas not very useful and we might come up with other techniques that might be more relevant.

- Parallel Decomposition
- Sequential Decomposition
- Input Space Reduction
- Duality
- Empirical Profiling
- Improve Search Guidance / Speculative Execution
- Workflow Feedback
- Incremental SAT Solving

## 6 Risks & Technical Feasibility

One of the risks of this project is that we might not be able find ideal use cases to test our optimization ideas in which they apply. Because different optimality problems have various possible routes for optimization it could both be difficult to find the perfect optimization scheme, that addresses most models as well as finding models that are ideal for our algorithms. This is why we are working with multiple collaborators to increase our problem space. That includes the graduate students in WATFORM who work on solving relevant problem sets and also Prof. Bryan Tolson from Department of Civil Engineering who has multiple problems that might be more helpful to solve using MOO.

Another unlikely risk for this project is that none of our algorithms provide satisfactory results in terms of performance benchmark, in which case it would still have research value however, little research impact since there would not be a lot of benefit to users from this.

Time is always a risk in research project such as this, because we might not be able to explore all our optimization ideas. Therefore we are paying close attention to which algorithms we want to explore first on the basis of how effective they might seem from initial analysis.

Table 1: Total Cost Breakdown

| Item | Cost |
|---|---|
| This is item #1 | $ABCDEF.01 |

# 7 Costs

```
//We estimate that it will cost around \$AB,CDE to pay the developers.
//This estimate is based on the average co-op student hourly earnings
//information~\cite{ref:ceca} provided by the University of Waterloo,
//and under the assumption that the entire project will require about
//ABCD man-hours to complete.
```

Aside from the cost associated with the labour, we estimate our expense to be around $A,BCD.EF as a result of purchasing various hardware components and/or software licenses. Table 1 shows the total cost breakdown for our project.

# 8 Legal/Social/Ethical Issues

We may face various legal issues with incorrect or suboptimal solutions generated by our solver. The use of these solutions without any verification done by our customer may result in undesired consequences, which include, but not limited to, bodily harm, violation of regulations, and monetary damages. As such, a disclaimer must be presented to our customers, advising them that verifications be done on all solutions and that we hold no liability in case of such [**insert your favourite word here**].

Moreover, by increasing the optimality of our solver, we may be able to solve MOO problems that were previously considered infeasible to be solved by machines. This will likely cause several ethical and social issues, the most predominant issue being the elimination of job positions that were previously tasked to solve those problems manually.

# References

[1] NASA. (2011). Decadal Survey, [Online]. Available: `http://science.nasa.gov/earth-science/decadal-surveys/` (visited on 11/25/2012).

[2] D. Rayside, H.-C. Estler, and D. Jackson, "The Guided Improvement Algorithm for Exact, General-Purpose, Many-Objective Combinatorial Optimization," MIT Computer Science and Artificial Intelligence Laboratory, Tech. Rep. MIT-CSAIL-TR-2009-033, Jul. 2009.

[3] M. Gavanelli, "An Algorithm for Multi-Criteria Optimization in CSPs," in *Proceedings of the 15th European Conference on Artificial Intelligence*, F. van Harmelen, Ed., Lyon, France: IOS Press, 2002, pp. 136–140.

[4] M. Lukasiewycz, M. Glaß, C. Haubelt, and J. Teich, "Solving multi-objective pseudo-boolean problems," in *Proceedings of the 10th international conference on Theory and applications of satisfiability testing*, ser. SAT'07, Lisbon, Portugal: Springer-Verlag, 2007, pp. 56–69.

[5] C. Dhaenens, J. Lemesre, and E.-G. Talbi, "K-PPM: A new exact method to solve multi-objective combinatorial optimization problems," *European Journal of Operational Research*, vol. 200, no. 1, pp. 45–53, 2010.

[6] R. Olaechea, S. Stewart, K. Czarnecki, and D. Rayside, "Modeling and multi-objective optimization of quality attributes in variability-rich software," in *International Workshop on Non-functional System Properties in Domain Specific Modeling Languages (NFPinDSML'12)*, Innsbruck, Austria, October 2012.