Optimizing and Visualizing the Allocation of Corporate Funding

Jacques Heunis
HNSJAC003
University of Cape Town

Timothy Gwynn
GWYTIM001
University of Cape Town

CCS Concepts

•**Mathematics of computing** → **Optimization with randomized search heuristics; Evolutionary algorithms;** *Bio-inspired optimization;* •**Theory of computation** → *Scheduling algorithms;* •**Computing methodologies** → *Genetic algorithms;* •**Human-centered computing** → **Visualization;** *HCI design and evaluation methods; Visual analytics; Visualization design and evaluation methods;* • **Information systems** → Enterprise resource planning;

Keywords

Optimization, Visualization, Genetic Algorithm, Particle Swarm Optimization, User-centred Design

1. PROJECT DESCRIPTION

Many large companies are composed of a number of smaller sub-companies, each with their own levels of income and expense. These sub-companies can be categorized by their profitability, either generating excess income (a funding source) or being in need of outside income (a funding sink). Ordinarily, companies that require extra income would need to request funding from an external source such as a bank. Being part of a larger company, however, allows funds to be easily loaned between sources and sinks. Currently, sources are matched to sinks manually by a company employee. Given the high level of complexity of the problem and the large space of possible solutions, manually created matches are liable to be sub-optimal. Having sub-optimal matches increases the total amount of funding required to satisfy all requirements, causing the company to rely on third-party funding which is provided at higher interest rates, costing the company money. In addition, manually matching sources to sinks requires a significant amount of employee time which could otherwise be spent gaining value for the company elsewhere.

Our project aims to find solutions to this problem for Old Mutual specifically, although it is reasonable to assume that our methods will be applicable in other contexts. To do this we will use two different methods to generate optimized solutions to the fund matching problem. Since there may be a number of similar solutions to the optimization problem we will also create a visualization designed to display the solutions visually and allow for different solutions to be compared to one another.

1.1 Optimization

1.1.1 *Problem Overview*

We have the following optimization problem: Given a set of funding sources and sinks, each with a starting date, duration, funding amount and tax class, match the sources to sinks, such that the total cost of satisfying all sinks is minimized. One popular way of approaching optimization problems in general is the use of stochastic search algorithms which avoid exponential computation time by adding an amount of randomness to their selection of what parts of the solution space to search through. This project will take the approach of applying stochastic search algorithms because they can be run on standard computer hardware in a reasonable amount of time and have been successful at solving similar problems in the past. More specifically, we will explore the use of Evolutionary Algorithms (EAs) which are stochastic search algorithms inspired by natural evolution.

One of the algorithms we will use is based on the Genetic Algorithm (GA) which was first proposed by Holland in 1975[7]. GA consists of a population (a set of solutions), mutation and crossover operators (unary and binary operations respectively which modify members of the population to produce new ones, a so-called "child" population), and a selection process (by which members of the current and child populations are selected for the next iteration's population). The selection process is based on a fitness function which evaluates candidate solutions (members of the population) to determine how good of a solution they are. Each successive population is called a "generation".

The other meta-heuristic algorithm we will use is an extension of Particle Swarm Optimization (PSO) which was first presented by Eberhart and Kennedy in 1995[5]. As with GA above it works by iteratively updating a set of solutions that converge over time. However, PSO is not generational. In PSO the population does not get replaced, only updated. Each iteration every member of the population considers the best solution encountered so far by itself, its neighbours (a set of randomly selected members of the population) and by the entire population. It then moves a small distance towards a weighted average of those three. This type of movement requires that moves of any magnitude and direction are possible in solution space, meaning that PSO is only applicable to problems with a continuous domain.

These algorithms are effective when optimizing a single function, however in our case we might want to take into account other factors such as tax class in addition to the amount of money required by each sink. Multi-objective Evolutionary Algorithms (MOEAs) are evolutionary algorithms that aim to simultaneously maximize or minimize more than one objective function. These objectives may be at odds with each other so finding a single best solution is usually not possible to do algorithmically. In order to be able to reason about the trade-offs between objectives and what it means for a candidate solution to be "optimal" the notion of Pareto dominance is used. A solution X is said to dominate solution Y if X is at least as good as Y with respect to every objective, but strictly better with respect to at least one objective. A solution is said to be non-dominated if there are no solutions which dominate it. The Pareto front is the set of all non-dominated solutions to a multi-objective problem, and this is what MOEAs aim to compute. Since the algorithm cannot say that one non-dominated solution is "better" than another, it gives all of them and leaves it up to the user to decide which one to use. Another concept which is important in some MOEAs is that of Pareto rank. Pareto rank is defined recursively with solutions on the Pareto front being given rank 1. If the solutions on the Pareto front were removed from the population and the front were recalculated, those solutions in the new front would have rank 2. This continues until the entire population has been assigned a rank.

1.1.2 Brief Description of Algorithms

We will test two algorithms, the updated Non-Dominated Sorting Genetic Algorithm (NSGA-II) by Deb et al.[3] and the Elitist-Mutation Multi-Objective Particle Swarm Optimization algorithm (EM-MOPSO) by Kumar et al[10]. These algorithms were selected based on their success, diversity, and handling of constraints. NSGA-II has been successfully applied to many problems in the literature and many new algorithm proposals (including Kumar's paper on EM-MOPSO) include a comparison to NSGA-II as a benchmark. EM-MOPSO is a more recently proposed algorithm but shows promise where it has been applied (Kumar's implementation was more effective than NSGA-II on his specific problem). Both algorithms have the benefit of including constraint-handling techniques in their specification. This removes the need for modification of the algorithms to include independently suggested means of handling constraints. These two meta-heuristics were selected as an exploration of approaches to the fund matching problem. Evaluating both GA- and PSO-based algorithms gives a broader look at what methods can be used to efficiently solve the problem.

NSGA-II is a well-known MOEA that is able to find good solutions to a variety of problems. Being a GA, NSGA-II is generational, but in addition to the usual population for every generation, an archive of the best solutions found so far in the entire run (IE across generations) is kept, and new generations are created from this archive population as well as the current generational population. This introduces so-called "elitism" which means that the algorithm favours known good solutions over the potential of unknown solutions. Every iteration the Pareto rank of every individual in the entire (generation and archival) population is computed and a new archive is created, adding individuals of highest rank until it is full. The output is the Pareto front of the last generation. NSGA-II improves on the original NSGA algorithm by introducing a more elitist selection procedure, removing a parameter that controlled the diversity of candidate solutions (thereby reducing the amount of tuning necessary to get good solutions), and by introducing a faster algorithm for sorting the population by Pareto rank.

EM-MOPSO is an extension of PSO to multi-objective problems. EM-MOPSO also uses an archive of good solutions which has a variable size but when it grows past a preset maximum size, solutions that are close to many other solutions in the archive are discarded. This ensures that the algorithm maintains a diverse set of good solutions. The algorithm gets its name from its elitist mutation operator which selects the least fit particles for one objective and replaces them with a mutation of a random solution from the archive which is in a less-explored area of the search space.

1.2 Visualization

1.2.1 Data interpretation

The data generated by the optimization needs to be presented in way that is simple to understand while also retaining all essential information. This is a challenge as the data consists of on the order of a 100 source and sink nodes, representing sources of funding or sinks requiring additional funding, each with multiple attributes. To display all this data at once especially in a purely textual format is ineffective and creates unnecessary work for the users of the data. Further complicating the problem is the fact that the data changes over time as sources and sinks become active and then expire. Therefore we aim to create an interactive visualization which will allow users to interrogate the data. There are a number of challenges with this goal. These include; creating an effective way to show how the data changes over time, allowing data sets from different time periods to be compared and choosing a format for the data to be displayed that balances the level of detail displayed and the simplicity of the display.

1.2.2 User-Centered design (UCD)

In order to ensure that the final product of the visualization aspect of this project is useful it is necessary to incorporate UCD into the process. This is because a creator's intuition is often very different from the user's. It is hard to accurately predict how users will intuitively try to use the product, especially in situations such as this where the designer is designing for a field outside his or her realm of knowledge. By focusing on UCD we put the user's needs first and then design the product to fit these.[9] Tory et al. explore the value of expert reviews in their 2005 paper *Evaluating visualizations: do expert reviews work?*.[11] They found that evaluation of visualizations by expert end users resulted in useful feedback that could be used to improve visual designs.

This means that the visualization will be designed together with the end users so that it allows them to optimally fulfil their goals while minimising frustrations they might have with the new tool. It entails getting regular feedback from users at each stage of the design process. This presents two major challenges. Firstly, arranging sessions in which users can test the design and provide feedback will require planning and effective communication between us and the end users at Old Mutual. Secondly, interpreting user feedback and using it to effectively improve the visualization tool while retaining control over the final design may prove to be challenging.

2. PROBLEM STATEMENT

Consider a set of funding sources and sinks, each with a start date, duration, funding amount, tax class and (in the case of sources) an interest rate. How can we find a matching of sources to sinks, such that the total cost of satisfying all sinks is minimized. In addition, how do you present a solution, or set of solutions to the matching problem in a visual way that most effectively enables users to make informed decisions about how to allocate funding.

2.1 Optimization Research Question

Can we find an meta-heuristic search algorithm solution that performs better than the existing manual method for the funding allocation problem?

Does the Non-dominated Sorting Genetic Algorithm, NSGA-II (as proposed by Deb et al.[3]), perform better than the existing manual method for the funding allocation problem?

Does the Elitist Mutation Multi-Objective Particle Swarm Optimization algorithm, EM-MOPSO (as proposed by Kumar et al.[10]), perform better than the existing manual method for the funding allocation problem?

2.2 Visualization Aim

To design a visualization of the funding allocation solution for Old Mutual that allows users to answer all the following visual queries related to the data. The visualization is also required to be able to show how the solution changes over time as well as facilitating the comparison of different solutions.

Visual Queries:

- What is the size (value) of a specific source?
- What is the size (value) of a specific requirement?
- What is the size (value) of a specific loan?
- Which sources are providing loans to a specific requirement?
- Which requirements are being provided with loans from a specific source or Balance pool?
- What is the relative interest rate offered by each source?

These queries, supplied by Old Mutual, may be refined and added to as more feedback is provided by users.

3. APPROACH

Because the optimization and visualization sections of this project are distinct from each other and are not interdependent we have taken a split approach. This allows for work on the optimization and visualization sections of the project to be split up effectively and carried out independently. Work on optimization will involve the implementation and comparison of genetic algorithms while the design of the visualization system will be carried out independently using UCD methods.

3.1 Optimization

In order to determine if either of the algorithms mentioned in Section 1.1.2 can be more effective than the current manual method, we will first create our own implementation of each of them. We will then test those implementations on the data provided by Old Mutual, tweaking algorithm parameters as necessary. In order to check that the output is statistically valid, statistical tests will be done for each algorithm to compare its output to the manually created solution.

3.2 Visualization approach

The basis of the interactive visualization will be a node-link graph. This was chosen as it can effectively display each source and sink node as well as the connections between nodes. This allows the primary visual queries defined by Old Mutual end users to be effectively addressed. Nodes will have visual indicators of their individual properties such as value and interest rate. Directed links will connect nodes to show how funds are being distributed as well as incorporating a visual indication of the amount being transferred through the link.

Since the solution changes over time it will be necessary to incorporate a dynamic element into the display of the node-link graph. The precise manner in which to do this will depend on user feedback as well as what is supported by the web tools being used to implement the visualization. From the review of literature on the subject the most likely methods are; animation between time steps, overlaying time steps or displaying separate time steps simultaneously but offset either vertically or horizontally.[1]

Additional visual elements will be necessary to display aggregate data. Such as the total unused amount in all sources. These may either be integrated into the main visualization or displayed separately as an overview window.

Additional functionality may be added to the visualization should it be practical to do so. An example of this would be the ability to track a specific source or sink node over multiple time steps.

4. PROCEDURES AND METHODS

The procedures and methods to carry out the two parts of this project are considerably different to one another. The optimization section will require the implementation of state of the art algorithms which will be a complicated and challenging procedure. The testing and evaluation of this section of the project is comparatively simple. The visualization section is based around a series of evaluations and will involve a number of design cycles. While these may not individually be as complex as the algorithm implementation they will require rapid design and a more complex set of deadlines to meet.

4.1 Optimization

4.1.1 *Implementation*

While an official implementation of NSGA-II is available on the internet[1], we could not find any comparable implementation of EM-MOPSO, and so we will instead create our own implementation for both algorithms. This is to ensure that the implementation that we use can accept data in the correct format. Using our own implementations also ensures that the results for each algorithm are comparable in terms of run time, which we can take into account when comparing the two algorithms against each other.

We will write our implementations in C++ because of the inherent speed and memory usage advantages that it holds over other modern languages such as C# or Python.

[1]Kanpur Genetic Algorithms Laboratory @ http://www.iitk.ac.in/kangal/codes.shtml

4.1.2 Parameter Tuning

Both algorithms have parameters that need to be specified ahead of time. In the case of NSGA-II they are the mutation and crossover operators and in the case of EM-MOPSO, they are the weights for each of the best-so-far solutions (self, neighbour, and population). There is no known procedure for finding optimal values for EA parameters and so we will simply try a variety of values for each and use the ones that result in the best output.

4.1.3 Run on real-world data

In order to accurately evaluate the effectiveness of the two algorithms, they will be run on real-world test data. This will be provided by Old Mutual and is actual data from the manual fund matching process with identifying or sensitive information removed. Using real-world data allows us to directly compare the output of our algorithm to the allocation created manually by an expert.

4.1.4 Test for statistical significance

Given the stochastic nature of evolutionary algorithms, a single run of each algorithm that produces a better result than the manual method does not necessarily imply that the algorithms are a more effective method in general. We will do statistical tests that compare the output produced by the algorithms to the output produced manually in order to see if one is statistically more effective than the other.

4.2 Visualization

4.2.1 User-centered Design (UCD)

Incorporating UCD into the design process will involve multiple stages of user input in different contexts. This will require a number of feedback sessions as well as artefacts for the users to base feedback on. To support this a number of prototypes, questionnaires and interview questions will need to be created. It will also be necessary to recruit a number of expert users in order to carry out the evaluation of the various design stages and provide input on additional functionality to be added to the design.

We aim to use at least three expert users in all stages our design process. They will be recruited from the treasury department at Old Mutual who will be the primary users of the final visualization product. The project is co-supervised by Old Mutual employees with direct access to members of their treasury department. It should therefore be possible to organize a number of feedback sessions through this connection. The input sessions from expert users will include an initial data collection interview and number of evaluations detailed below. Evaluations will either be carried out in person or online for the web-based prototypes.

4.2.2 Initial data collection

A short interview with a representative of the Old Mutual treasury team will allow for initial data collection. The subjects of this interview will be the primary visual queries that end users will need to address, the relative importance of various data dimensions and the scale of the data. The information gained will be used to create the initial design concepts of the visualization.

4.2.3 Prototyping

Initially a paper prototype or very basic implementation with extremely limited interactivity will be created. This will be used to ascertain that all necessary information is being displayed by the core design. It will also be used to determine how comfortable users are with the design.

Two Web-based prototypes will be created. The first will focus only on core features and include limited interactivity. It will be used to check that the concepts described by the paper prototype can be transferred onto an online medium. The second web-based prototype will contain all static visualization features but will still not introduce any functional interactivity. It will be used to check that all visual elements are present and satisfactorily displayed prior to interactivity being introduced to the design.

4.2.4 Interactive Versions

The initial interactive design will be made to work with a specific data set. It will be used to demonstrate the potential of an interactive visualization as well as give an almost complete impression of how the final visualization will function.

The second interactive visualization will update both visual and interactive elements of the initial interactive version. It will also be designed such that it is sufficiently robust to handle any valid data as input. This will be the penultimate design and only small bugs and necessary changes are expected to be found at this point.

The final design will correct any problems received from feedback on the previous design or independently noticed by the designer. Large changes are extremely unlikely at this point. The evaluation of this design will be more formal and detailed than in previous designs as it will be used to measure the success of this project. The evaluation will consist of a number of tasks which simulate common use cases for the visualization and a semi-structured interview.

5. ETHICAL, PROFESSIONAL AND LEGAL ISSUES

5.1 Anonymization of Data

Given that we are working with financial data from a large company, it is important to consider what information both our input and output data contains. In particular it should not be possible to identify any specific person or company from the data. Since we do not collect the data ourselves, it is left to Old Mutual to provide us with data that has already had any identifying information removed.

5.2 User testing

There are some ethical issues related to using users to test the visualization design. To address these we will need to go through the standard university procedures for utilizing human resources in our project. Due to the lack of any known risks for the users during the testing process this should be possible. Additionally we will need the consent of the users to participate in our study. Since the users have motivation to provide input to the design this should also be relatively simple.

5.3 Patents

To the best of our knowledge, the algorithms that we will implement (NSGA-II and EM-MOPSO) are not currently patented. This means that there are no legal issues involved in implementing and using them for this project.

6. RELATED WORK

6.1 Optimization

MOEAs can be applied to a wide variety of tasks including planning, scheduling, packing, circuit design and image processing[13]. While no work has yet been done on applying multi-objective evolutionary algorithms to funding allocation, literature that presents new algorithms or methods for solving similar problems can used as a source of direct solutions or techniques that can be individually applied. For example Deb et al. presented a method for handling constraints in their original paper on NSGA-II[3], however, Yen et al. later proposed a method of handling constraints that can be used with a variety of algorithms (including NSGA-II)[12]. Although we have chosen to evaluate only two algorithms, many general multi-objective optimization algorithms can be applied to funding allocation. Other algorithms that could be used include the Strength Pareto Evolutionary Algorithm (SPEA-2)[14], the Pareto Archived Evolution Strategy (PAES)[8] and the Pareto Envelope-based Selection Algorithm (PESA-2)[2].

6.2 Visualization

6.2.1 Node-link graphs in Finance

While Fund Managers are not exactly the same as fund matching they exhibit many of the same attributes. Namely they are made up of a time dependent network of objects that can be represented as a node-graph. Dwyer et al. created two variations of 3D graphs with the aim of visualizing the behaviours of Fund managers.[4] Although these visualizations are relatively simple they showcase the use of node-link graphs to model data similar in nature to the data being visualized in this project.

6.2.2 Dynamic Graphs

Dynamic graphs are used to represent the relationships between a number of entities and how these relationships change over time. The nature of the data in the fund match visualization is of sources and sinks being connected to one another for periods of time before disconnecting.

In their paper The State of the Art in Visualizing Dynamic Graphs Beck et al. explore how different variations of dynamic graph visualizations can be broken down and how the separate approaches differ from one another.[1] A number of the methods they discuss for displaying dynamic graphs such as Animation or super-imposition are promising options for this visualization project.

6.2.3 User-Centered design

User testing is an essential component of User-Centered design. In their paper on User-Centered design and evaluation Hix et al. break user testing into distinct categories; Heuristic evaluation, Formative evaluation and Summative evaluation.[6] Hix et al. carried out a number of rounds of evaluation of a realtime battlefield visualisation using these methods.Unfortunately the methods described in Hix et al.'s paper would need both time and resources beyond the scope of this project. However Tory et al. explore the value of a expert review process in their 2005 paper and found it to yield effective results.[11] Since this method is more likely within the scope of this project it will form the basis of our UCD method.

7. ANTICIPATED OUTCOMES

7.1 Optimization

7.1.1 Better fund matching

We anticipate that the algorithms mentioned in Section 1.1.2 will perform better than the existing manual method of funding allocation. A more efficient allocation requires less money to be loaned from third-parties, thereby saving money for Old Mutual.

7.1.2 Determining success

The success of the optimization methods will be determined by the results of the statistical tests. If either algorithm produces solutions that are better than the manual method with at least a 90% level of confidence, the project will be considered a success.

7.2 Visualization

7.2.1 Visualization system

This project will result in a visualization system that will effectively display optimized fund match data. This system will be fully interactive and will allow for the comparison of data sets whether from different time periods or using different optimization parameters.

7.2.2 Human data analysis efficiency

The visualization will primarily aid in human data analysis. It will enable users to quickly identify patterns in the data and evaluate the quality of the optimization. It will also allow a number of visual queries to be efficiently answered as well as allowing the user to interact with the data on different levels of detail.

7.2.3 Determining success

The success of the visualization system will be determined by the user evaluations of the final visualization. If the users can complete all the tasks, to be based on actual everyday tasks users need to perform, given to them within a reasonable period of time this will be a signifier of success. Additionally if the feedback from the semi-structured interview conducted after the tasks is positive, users express a desire to continue using this tool over previous tools, this will be an additional indicator of a successful design.

8. PROJECT PLAN

8.1 Risk Matrix

Figure 8.1 shows the risks that we have identified for the project along with their impact and strategies for dealing with them.

8.2 Timeline/ Gantt chart

Figure 8.2 shows the projected timeline for the projected, including all major milestones.

8.3 Resources

Risk Condition	Consequence	Category	Prob	Impact	Mitigation	Monitoring	Management
Unable to find appropriate encoding for optimization	Optimization will not be possible or results will be very poor.	Design	Low	Critical	Communicate with Old Mutual to understand the importance of the various factors for optimization.	Constant referral back to the project specification to ensure that the selected encoding is correct and appropriate.	Reduce the problem to a simpler version that can be more easily described.
Lack of Raw Data	Inability to properly evaluate the performance of the optimization algorithm	Development	Low	Critical	Gain access to raw data as quickly as possible. One set already acquired.	Communicate with Old Mutual regularly to ensure they are aware of this need.	Create Representative data sets to test the optimization algorithm with.
Lack of Solution data	Inability to test Visualization tools on realistic data sets.	Development	Low	Critical	Gain access to solution data as quickly as possible.	Communicate with Old Mutual regularly to ensure they are aware of this need.	Create Representative data sets or use the optimization algorithm to generate solutions using raw data.
Lack of expert user testers	Unable to incorporate user-centered design into visualization design process	Design	Low	Critical	Do testing as early and as often as possible, Allow for testing to be carried out online.	Communicate regularly with Old Mutual as to the availability of employees to carry out tests.	Use non-expert testing to evaluate general visualization design.
Underestimating the time taken to create prototypes/ design iterations	Work will fall behind schedule and user feedback sessions may need to be rescheduled	Planning	Medium	Medium	Allow for some slack time in plan to catch up on unforeseen problems in design iterations.	Monitor progress of each design iteration and make adjustments to schedule if necessary	Introduce measures to prevent further delays and increase work pace to catch up.
Absence of a team member	Work will fall behind schedule	Project Length	Medium	Medium -Low	Team members plan ahead to anticipate absences	Communication between team members.	Redistribute workload to take into account lost time.
Scope of project increases	Development time may increase.	Development	Low	High-Medium	Ensure scope of project is well defined.	Communicate with Old Mutual to ensure all parties agree on the scope	Drop least necessary features from current scope.

Figure 1: A matrix depicting the risks to the project and their potential impact.

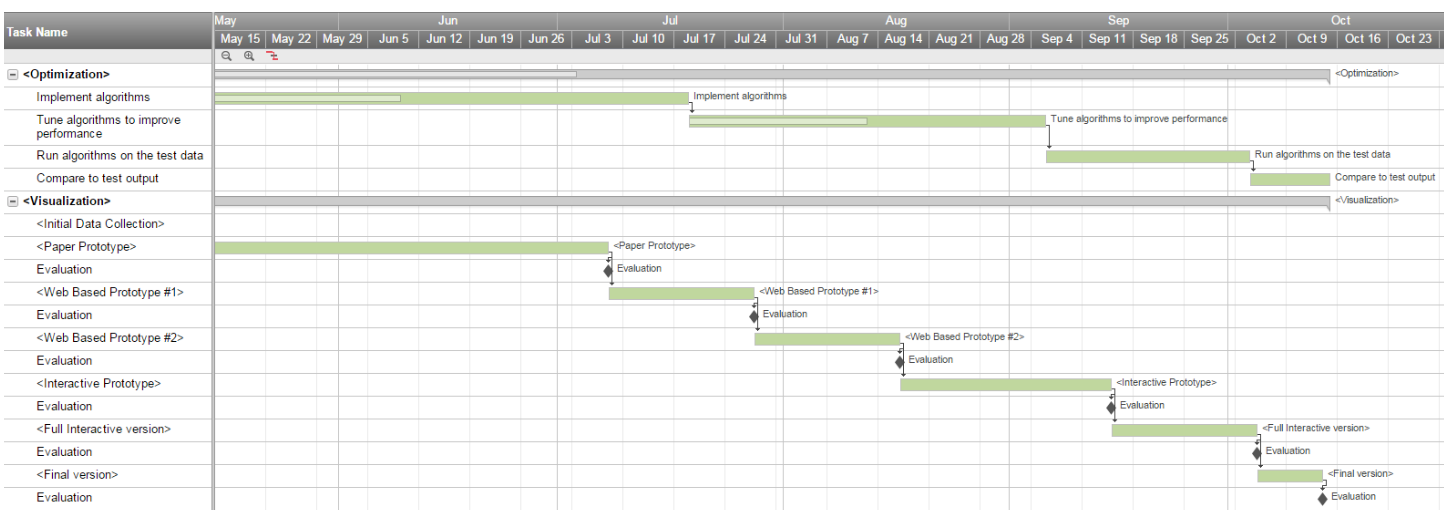

Figure 2: A matrix depicting the risks to the project and their potential impact.

8.3.1 Testers

We will require at least 3 expert user testers from Old Mutual for each design phase. These do not need to be consistent individuals between cycles. Testers should be part of the treasury department at Old Mutual, the targeted end users of the visualization. There will be 6 rounds of user testing, some online and some in person. Evaluations will take a maximum of 30 minutes per tester.

8.3.2 Hardware/Software

No specialized hardware or software is needed other than standard PC or laptop hardware and what is freely available on the internet.

8.4 Deliverables

8.4.1 Algorithm Implementations

- Initial implementation of both algorithms
- Tuned implementations for our specific problem.

8.4.2 Optimization results on example data sets

- Results of running each algorithm
- Statistical test results for comparison of methods

8.4.3 Visualization Systems

- Paper prototype.
- Semi functional web based prototypes.
- Full Interactive Visualization implementation

8.4.4 User test materials

- Initial usability data collected.
- Prototype feedback/ questionnaires.
- Formal task list for evaluation of fully interactive visualization.
- Final feedback interview transcript.

8.4.5 Project

- Project proposal presentation
- Draft of Final report
- Final Report
- Project Poster
- Project Website
- Reflection paper on the project

8.5 Milestones

8.5.1 Optimization Milestones

1. **Implement algorithms**
   Create a new implementation of each algorithm that takes input and gives output in an appropriate format. This format is important as the visualization will expect a specific input data format.
2. **Tune algorithms to improve performance**
   Find values for each parameter that make each algorithm most effective at solving our problem.
3. **Run both algorithms with the test data**
   Run the algorithms on Old Mutual's input data in order to get allocations that are comparable to Old Mutual's manually created output data.
4. **Compare to test output**
   Compare our results to Old Mutual's results to see if the algorithms produce more efficient allocations than the manual method.

8.5.2 Visualization Milestones

1. **Identify Visual queries**
   Perform initial interview with end users to identify core visual queries the visualization should be able to answer.
2. **Acquire Solution Data Set**
   Receive at least one example set of data to be visualised. This combined with the initial visual queries will be used to design the initial visualization concept.
3. **Paper Prototype**
   Create a paper prototype of the visualization concept. Test and evaluate this prototype with expert users. This evaluation will need to be done in person as the observer is required to create the interactivity.
4. **Web Prototype #1**
   Create a Web Based prototype of the visualization concept. Test and evaluate this prototype with expert users. This prototype will have limited interactivity and core visualization features. Evaluation may be performed online or in person.
5. **Web Prototype #2**
   Improve Web Based prototype of the visualization concept. Test and evaluate this prototype with expert users. This prototype will have limited interactivity but will implement all non-interactive visualization features. Evaluation may be performed online or in person.
6. **Interactive Prototype**
   Create a Fully functional prototype of the visualization concept. Test and evaluate this prototype with expert users. This prototype will have full interactivity and core visualization features. May only work on specific data sets. Evaluation should be performed in person.
7. **Fully Functional Visualization** Create a Fully functional Visualization system. Test and evaluate this system with expert users. This system will have full interactivity and all visualization features. Should work on any valid data set. Evaluation should be performed in person.

8. **Final Visualization**
   Update the visualization system to accommodate feedback. This is the final version of the design.

9. **Evaluate Final Visualization**
   Perform evaluation of the final visualization system using a series of tasks as well a semi-structured interview

8.6 Work Allocation

The project has two clear and distinct components: Optimization of the funding allocation, and visualization of the solutions to the optimization problem. The work will be split with Jacques handling the optimization while Tim deals with the visualization component.

9. REFERENCES

[1] F. Beck, M. Burch, S. Diehl, and D. Weiskopf. The state of the art in visualizing dynamic graphs. *EuroVis STAR*, 2014.

[2] D. W. Corne, N. R. Jerram, J. D. Knowles, M. J. Oates, and M. J. Pesa-ii: Region-based selection in evolutionary multiobjective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 283–290. Morgan Kaufmann Publishers, 2001.

[3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[4] T. Dwyer and P. Eades. Visualising a fund manager flow graph with columns and worms. In *Information Visualisation, 2002. Proceedings. Sixth International Conference on*, pages 147–152. IEEE, 2002.

[5] R. C. Eberhart, J. Kennedy, et al. A new optimizer using particle swarm theory. In *Proceedings of the sixth international symposium on micro machine and human science*, volume 1, pages 39–43, 1995.

[6] D. Hix, J. E. Swan, J. L. Gabbard, M. McGee, J. Durbin, T. King, et al. User-centered design and evaluation of a real-time battlefield visualization virtual environment. In *Virtual Reality, 1999. Proceedings., IEEE*, pages 96–103. IEEE, 1999.

[7] J. H. Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* U Michigan Press, 1975.

[8] J. Knowles and D. Corne. The pareto archived evolution strategy: a new baseline algorithm for pareto multiobjective optimisation. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 1, page 105, 1999.

[9] D. A. Norman. *The design of everyday things: Revised and expanded edition.* Basic books, 2013.

[10] M. J. Reddy and D. Nagesh Kumar. Multi-objective particle swarm optimization for generating optimal trade-offs in reservoir operation. *Hydrological Processes*, 21(21):2897–2909, 2007.

[11] M. Tory and T. Möller. Evaluating visualizations: do expert reviews work? *Computer Graphics and Applications, IEEE*, 25(5):8–11, 2005.

[12] Y. G. Woldesenbet, G. G. Yen, and B. G. Tessema. Constraint handling in multiobjective evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 13(3):514–525, 2009.

[13] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32 – 49, 2011.

[14] E. Zitzler, M. Laumanns, and L. Thiele. Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, pages 95–100, 2001.