Evolufy: A Browser-based Investment Portfolio Optimization Library Utilizing Genetic Algorithms and Transfomers for Sustainable Decision-making

Carlos Eduardo Sánchez Torres

Universidad Autónoma de Baja California

Faculty of Science

Bachelor of Computer Science

In partial fulfillment of the requirements for the Degree of Bachelor of Computer Science.

Author Note

Carlos Eduardo Sánchez Torres https://orcid.org/0000-0001-5799-4067

Correspondence concerning this article should be addressed to Carlos Eduardo Sánchez Torres, Faculty of science, Universidad Autónoma de Baja California, Carretera Transpeninsular 3917, Playitas, 22860 Ensenada, B.C. Email: a361075@uabc.edu.mx

Abstract

We propose the development of a WebAssembly library for investment portfolio selection involving Mexican shares, with the aim of democratizing data-driven investment decisions through browser-based applications. This tool leverages the principles of Modern Portfolio Theory, an optimization model that balances risk and returns. Genetic algorithms are employed to efficiently navigate the vast search space of potential portfolio combinations. Meanwhile, transformer models enhance Markowitz's model through sentiment analysis of news data. This integration of machine learning techniques into portfolio optimization provides more insightful investment strategies by taking into account both historical trends and market sentiment.

Keywords: multi-objective problems, optimization algorithms, evolutionary computation, investment portfolio optimization, web

Evolufy: A Browser-based Investment Portfolio Optimization Library Utilizing Genetic Algorithms and Transfomers for Sustainable Decision-making

Portfolio optimization is a crucial task in investment management, where the perfect balance between maximizing returns and minimizing the risk of investor users is sought. The complexity of this problem increases exponentially when considering several constraints and multiple objectives Coello et al. (2006). Moreover, the size of the search space can be enormous, especially in modern finance, where investors can choose from thousands of assets. To solve this problem, several algorithms have been developed, primarily in academia Coello et al. (2006) . However, few of these algorithms are available on the browser and can be used by investors and stock market speculators. Also, many of these algorithms do not consider realistic constraints, which can lead to unfeasible solutions in practice (Aranha & Iba, 2008; Liagkouras & Metaxiotis, 2015; Turcas et al., 2017). In this work, we propose to develop a software as a service and an open-source library for portfolio optimization, which will run on the browser and consider realistic constraints Apipie and Georgescu (2019) .

I'd chosen C++20, clang and Rational Unified Process to solve a constrained investment portfolio optimization problem (CPOP) from the state-of-the-art and compare my API suggesting improvements to the algorithms provided by Coello et al. (2006) bringing into play (Aranha & Iba, 2008; Liagkouras & Metaxiotis, 2015; Turcas et al., 2017) assessing by Apipie and Georgescu (2019) metrics e.g. we'll select same 20 Bucharest Stock Exchange assets and after 500 stocks from S&P500.

So, it's an applied project driven by a quantitative method whose intended readers are developers in the finance industry, investors, stock market speculator, managers, and other researchers concerned about portfolio optimization.

In following sections, you'll read about how to choose a portfolio on Modern Portfolio Theory (MPT) introduced by (Markowitz, 1952; Roy, 1952) and find it a solution on Multiobjective Evolutionary Algorithm (MOEA), and constructing a solution with

C++20 who works in the browsers from scratch without servers.

Problem Description and Statement

Allocate a fixed amount of capital among n assets such as stocks, funds, bonds, and so forth in order to find the best Sharpe Ratio: maximizing the expected return and minimizing the risk -the covariance matrix of returns- from recent data in the browsers. So, it's constrained multi-objective optimization.

We hypothesize that a software-as-a-service and an open-source library that operates solely on a browser, drawing upon evolutionary algorithms and Modern Portfolio Theory, can provide an efficient and universally accessible solution for multi-objective constrained portfolio optimization. We anticipate that this approach will outperform current methods in terms of efficiency and feasibility, thereby offering end-users a superior tool for investment management.

Goals

The project aims at constructing an open-source software library in C++20 running in the browser for constrained multi-objective optimization by new evolutionary algorithms based on Coello's work to solve a Markowitz's portfolio optimization problem. It includes objectives like implementing the algorithms, benchmarking the results obtained, and suggesting improvements in space, time, and clarity about them.

For that, we propose the below specific goals:

1. Develop and program evolutionary algorithms for investment portfolio optimization, based on the works of Coello and other relevant authors.

2. Implement a web interface for using the library, allowing users to input their own constraints and objectives.

3. Conduct tests and performance analysis of the implemented algorithms, using real-time stock market data.

4. Compare the performance of our library with other existing portfolio optimization software packages.

5. Provide clear documentation and tutorials to facilitate the use and future development of the library.

Why matters?

Financial institutions are relentless maximize their returns at an up-to-scratch level of risk, so choosing the best portfolio is an essential asset of fund management.

Since Sefiane and Benbouziane (2012) confirms evolutionary algorithms' validity and efficiency, better techniques will have better-applied results as pretends to do this project.

Background

Although the industry has unlike solver packages to optimize a portfolio involving evolutionary algorithms, quadratic programming, and so forth, none runs in the Web from a client (table 1 summarizes the packages). So the leading knowledge fields are evolutionary algorithms in the Web browsers in order to choose an investment portfolio in real time.

We know which are the most frequently used multiobjective evolutionary algorithm (MOEAs) and their performance metrics for solving the portfolio optimization problem by (Apipie & Georgescu, 2019; Coello et al., 2006; Liagkouras & Metaxiotis, 2015). See table 2.

Because realistic constraints become CPOP to a quadratic mixed-integer problem (QMIP) that is discrete NP-hard (Liagkouras & Metaxiotis, 2015), even though "there ain't no such thing as a free lunch" exists (Wolpert & Macready, 1997) choosing a right algorithm in order to perform better than each other (Apipie & Georgescu, 2019) is critical.

Theory

Optimization and Genetic Algorithms

Optimization dates back millennia, from Plato (427-347 BCE) and Aristotle (384-322 BCE) used to find the best society to most recently it used to allocate resources in World War II with George Dantzig (1914-2005). We'll focus on multi-objective optimization problems in a numerical sense and ending up analyzing a solution approach —genetic algorithms.

First off, we deal with present the basic optimization problem that is

$$\text{minimize } f(\mathbf{x}) \tag{1}$$

$$\text{subject to } \mathbf{x} \in \chi \tag{2}$$

where x is a design point, χ is the feasible set and it can be discrete or continuous, and f is the objective function. Among all points in the feasible set χ, the $\mathbf{x}$ that minimizes the objective function is called a solution or minimizer. A particular solution is denoted as $\mathbf{x}^*$. As we work with n-dimensional space, x is a vector and it's written as usual:

$$\mathbf{x} = [x_1, x_2, x_3, ..., x_i, ..., x_n] \tag{3}$$

where x_i is called decision variable or design variable.

Easily a minimization problem can be replaced by maximization problem and vice versa. From

$$\text{maximize } f(\mathbf{x}) \text{ subject to } \mathbf{x} \in \chi \tag{4}$$

to

$$\text{minimize } -f(\mathbf{x}) \text{ subject to } \mathbf{x} \in \chi \tag{5}$$

Kochenderfer and Wheeler (2019)

But, what is a feasible set? A feasible set is defined by the whole constraints where each constraint restricts of possible solutions. They are written with $\leq$, $\geq$, or $=$ since we work with numerical optimization. So, our problem follows the General Multiobjective Optimization Problem definition by Coello et al. (2006) that is

$$F(\mathbf{x}) = (f_1(\mathbf{x}), ..., f_k(\mathbf{x})) \tag{6}$$

subject to

$$g_i(\mathbf{x}) \leq 0, i = 1, ..., p \tag{7}$$

$$h_j(\mathbf{x}) = 0, j = 1, ..., q \tag{8}$$

and our goal is to find global minimizer, a particular $\mathbf{x}^*$ such that $f(\mathbf{x}^*) \leq f(\mathbf{x})$.

But some multi-objective problems are a tradeoff between costs, performance, and time, and it is unclear how to prioritize constraints, so when we search them a solution, we use Pareto optimality that is a design of equilibrium between tradeoff objectives. Then, we will describe domination, a criterion that compares two solutions.

Domination's definition. Domination is a relationship such that $\mathbf{x}$ dominates $\mathbf{x}'$ iff $f_i(\mathbf{x}) \leq f_i(\mathbf{x}')$ for all $i \in \{1, ..., m\}$ and $f_i(\mathbf{x}) < f_i(\mathbf{x}')$ for some i.

Also we say $\mathbf{x}$ is better than $\mathbf{x}'$. The non-dominated set is one that no point dominates its elements in criterion space, which is called the Pareto frontier where criterion space is the image of χ through f, and it's written Υ. Criterion space is sometimes called objective function space.

Non-dominated set generator

A subproblem is how we can generate a non-dominated set but no criterion space in order to shed light on algorithm behavior. You can check out our Non dominated set generator CLI.

Simplex and monotonic decreasing functions are non-dominated set. For instance 1 2 3 4 5.

How can we solve multi-objective problems?

We can solve multi-objective problems with enumerative, deterministic, or stochastic techniques, where genetic algorithms are the last ones, those heuristic algorithms are our object of study. A non-exhaustive list can be found on 6 and MOEA types and their score to solve CPOP is on 2.

We measure technique's solution with performance metrics 3, but most popular is hypervolume. Hernández Gómez (2018) made a fast unpublished C framework to calculate those metrics.

A genetic algorithm is defined by Coello et al. (2006), in short, it's a bio-inspired algorithm using evolution as a metaphor. I show you an outline below.

A foundational concept is population that is the generalized composite data structure on evolutionary algorithms 7.

Algorithm genetic do the following with population:

INITIALIZE-POPULATION makes a random population from design space and encodes it to binary, real, permutation, or tree.

THE TERMINATION CRITERION IS NOT TRUE uses it to determine when the algorithm ends. Perhaps, the easiest way is a generation limit.

SELECT-PARENTS-FOR-THE-NEXT-GENERATION evaluates a decoding population on the objective function, after that procedure applies a fitness function, and then it makes a mating pool with parents by the canonical selection, roulette wheel selection, or other. When the procedure always pass the best parent is called elitism.

CROSSOVER combines parents from mate pool to make offsprings. Some crossover schemes are single-point crossover, two-point crossover, uniform crossover.

MUTATE allows new traits exploration. Some mutation schemes over offsprings are flipping each bit with small rate for bit-valued chromosomes and Gaussian mutation for real-valued chromosomes.

Worked example

A worked example is useful to understand optimization problems and genetic algorithms. In genetic algorithms with elitism, each generation is closer to global minimum than previous generations how you can appreciate on 8.

$$F(x, y) = min(x^2 + y^2) \tag{9}$$

subject to

$$-5.14 \leq x \leq 5.14 \tag{10}$$

$$-5.14 \leq y \leq 5.14 \tag{11}$$

You can check out a genetic algorithm implementation on https://gist.github.com/sanchezcarlosjr/dc500b87169f1f0be17158ecb376e377. Another worked example is the path finder problem on https://graph-theory.sanchezcarlosjr.com/.

On Portfolio Theory

A lot of people who want to be rich fast but those will discover the problems of that statement. We're going to follow the Graham's mantra: Speculators, Investors, Traders

Financial markets

Public and private markets. Cryptocurrencies.

Stocks

Stock issues

Intraday

Financial statements

Dividends

Indices

Rates

Commodities

Prices. We define an price of an stock issue as P_t at time t.

Returns. We define an return of an stock from time $t-1$ to time t as:

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}} \tag{12}$$

log-Returns

Mexican Stock Exchange

Data from an API to access mexican market. We've chosen Databursatil to get data from Mexican Stock Exchange.

Modern Portfolio Theory

In spite of (Buffett, 1997; Hathaway, 1996) said the MPT is twaddle and has no utility since its risk definition and its tendency to diversify are wrong incluiding it is precisely wrong than approximately right —value investing, MPT is the current paradigm.

Return. The expected returns of

The constrained portfolio optimization problem (CPOP) is formulated as

$$f_i : \Omega \to R^m, \Omega \subset R^m, \Omega : \text{search space} \tag{13}$$

$$\text{Optimize objetive functions } f(w.) = (f_1(w), f_2(w)) \tag{14}$$

$$\text{Maximize portfolio return } f_1(w) = \sum_{i=1}^{m} w_i r_i \tag{15}$$

$$\text{Minimize portfolio return } f_2(w) = \sum_{i=1}^{m} \sum_{j=1}^{m} w_i w_j \sigma_i \sigma_j \rho_{ij} \tag{16}$$

where

$$m \equiv \text{number of stocks}, i, j \in [1, m]. \tag{17}$$

$$w_i \equiv i \text{ decision variable, } w_i \subset \Omega. \tag{18}$$

$$r_i \equiv \text{rate of return of asset } i. \tag{19}$$

$$\rho_{ij} \equiv \text{correlation between assets } i,\ j \text{ such as } -1 \leq \rho_{ij} \leq 1. \tag{20}$$

$$\sigma_i, \sigma_j \equiv \text{standard deviation of stocks returns } i \text{ and } j. \tag{21}$$

subject to

$$\sum_{i=1}^{m} w_i = 1 \tag{22}$$

$$0 \leq w_i \leq 1 \tag{23}$$

Assumptions. MPT assumes the following:

- Investors are rational.
- Market is perfectly efficient.
- All investors have the same information.
- No taxes or transaction costs.

Roudier, 2007

WASM library on EvaNotebook

We've chosen safe, fast and portable browser development Sletten, 2021 because software no installation is required, the ecosystem is free of operating system, the most devices has one, and now it's possible run C++ language . Of course, we may be limited to local hardware capabilities.

WebAssembly are the assembly set instructions running in the web and it executes a particular chip's instructions. Hence, you may compile C++ using the clang with backend wasm-ld. We made a toy project as proof of concept —calculator intrepreter, it's on Calculator.

EvaNotebook is a P2P Browser-Based Computational Notebook for Real Systems with IA and an open-source platform designed to accelerate your AI development journey. Built for web developers and AI enthusiasts alike, this powerful notebook makes it easy to create and collaborate on projects in real time, directly from your browser.

Methods

It is an applied project in computer science which means implementing state-of-the-art software —their efficiency and accuracy are our objects of study. Reliability concerns the robustness of the construction —free of "bugs" and crashes, and since the Rational Unified Process guarantees it —Jacobson et al. (1999), I chose it. You can see RUP phases and disciplines to 9, hence we'll use UML and develop iteratively.

On the other hand, validity is the accuracy between our MOEA output and Apipie and Georgescu (2019) performance metrics and experimental design, which means 3.

We'll choose same 20 Bucharest Stock Exchange assets as Apipie and Georgescu (2019) and after 500 stocks from S&P500 with real time data provided by Yahoo Finance API.

Even though our method can be reliable and valid, it can be twaddle and don't reflect the best portfolio investment.

Schedule

Following the Rational Unified Process, this is the schedule 10.

Conclusion

Since my current results, we can deduce it is possible to run efficiently on web genetic algorithms in order to solve CPOP.

References

Apipie, F., & Georgescu, V. (2019). Assessing and comparing by specific metrics the performance of 15 multiobjective optimization metaheuristics when solving the portfolio optimization problem. *ECONOMIC COMPUTATION AND ECONOMIC CYBERNETICS STUDIES AND RESEARCH*, *53*, 39–58. https://doi.org/10.24818/18423264/53.3.19.03

Aranha, C. C., & Iba, H. (2008). A tree-based ga representation for the portfolio optimization problem. *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, 873–880. https://doi.org/10.1145/1389095.1389267

Buffett, W. E. (1997). Chairman's letter. *Annual shareholder letters*. https://www.berkshirehathaway.com/letters/1996.html

Coello, C. A. C., Lamont, G. B., & Veldhuizen, D. A. V. (2006). *Evolutionary algorithms for solving multi-objective problems (genetic and evolutionary computation)*. Springer-Verlag.

Hathaway, B. (1996). Afternoon session - 1996 meeting. https://buffett.cnbc.com/video/1996/05/06/afternoon-session---1996-berkshire-hathaway-annual-meeting.html

Hernández Gómez, R. (2018). Parallel hyper-heuristics for multi-objective optimization. https://repositorio.cinvestav.mx/handle/cinvestav/2349

Jacobson, I., Booch, G., & Rumbaugh, J. (1999). *The unified software development process (paperback)*. Addison-Wesley Educational.

Kochenderfer, M. J., & Wheeler, T. A. (2019). *Algorithms for optimization*. The MIT Press.

Liagkouras, K., & Metaxiotis, K. (2015). Efficient portfolio construction with the use of multiobjective evolutionary algorithms: Best practices and performance metrics. *International Journal of Information Technology & Decision Making*, *14*(03), 535–564.

Markowitz, H. (1952). Portfolio selection*. *The Journal of Finance*, *7*(1), 77–91. https://doi.org/https://doi.org/10.1111/j.1540-6261.1952.tb01525.x

Roudier, F. (2007). Portfolio optimization and genetic algorithms. *Master's thesis, Department of Management, Technology and Economics, Swiss Federal Institute of Technology (ETM), Zurich.*

Roy, A. D. (1952). Safety first and the holding of assets. *Econometrica: Journal of the econometric society*, 431–449.

Sefiane, S., & Benbouziane, M. (2012). Portfolio selection using genetic algorithm. *Journal of Applied Finance & Banking*, *2*, 143–154.

Sletten, B. (2021). *Webassembly: The definitive guide: Safe, fast, and portable code.* O'Reilly Media, Incorporated. https://www.oreilly.com/library/view/webassembly-the-definitive/9781492089834/

Turcas, F., Dumiter, F. C., Brezeanu, P., Farcas, P., & Coroiu, S. (2017). Practical aspects of portfolio selection and optimisation on the capital market. *Economic Research-Ekonomska Istraživanja*, *30*, 14–30. https://doi.org/10.1080/1331677X.2016.1265893

Wolpert, D., & Macready, W. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67–82. https://doi.org/10.1109/4235.585893

Table 1

Multi-objective optimization software packages

Name
Pymoo
MOSEK
Excel Solver Function
TOMLAB
NAG Numerical Library
Gurobi
KNITRO
CPLEX
SCIP
GLPK
Glpk.js (JavaScript)
Math.js (JavaScript)
Numeric JavaScript (JavaScript)
GlueNum (JavaScript)
Numerical JavaScript (JavaScript)
Genetic Algorithm JS (JavaScript)
Genetic.js (JavaScript)
synapsis.js (JavaScript)

Table 2

MOEA types and their score to solve CPOP

MOEA type	Research Percentage (%)	Score
MOPSO		70
NSGAII	29.17	47
SPEA2	25.00	51
PESA	16.67	
SPEA	8.33	
PAES	8.33	
MOGA	4.17	
NPGA2	4.17	
IBEA	4.17	
SPEA2SDE		56
MOEADDRA		54

Table 3

Performance metrics

Metric
Hypervolume (HV)
Generational distance (GD)
Inverted generational distance (IGD)
Averaged Hausdorff distance (Δp)
Spread metric Δ
Spacing metric (S)
Coverage of two sets (C)
Riesz s-energy

Figure 1

Cos

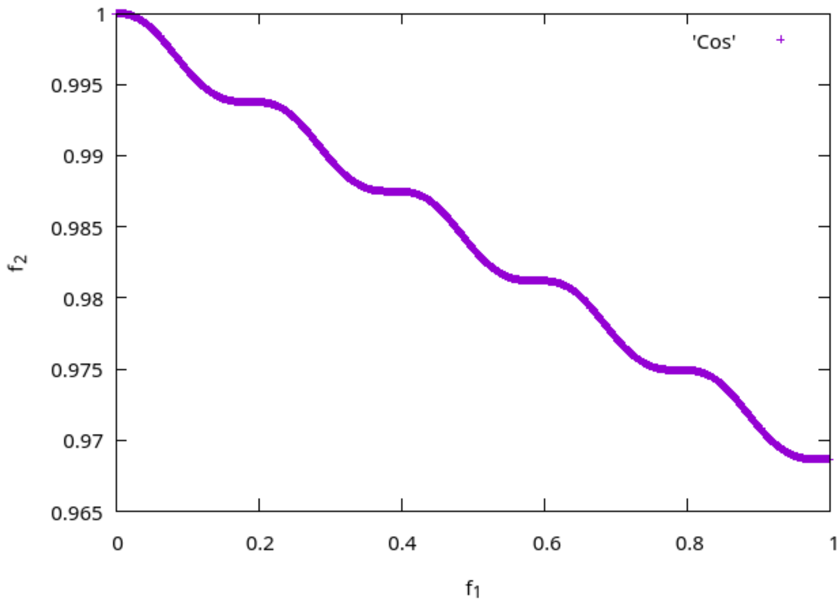

Figure 2

Exponential

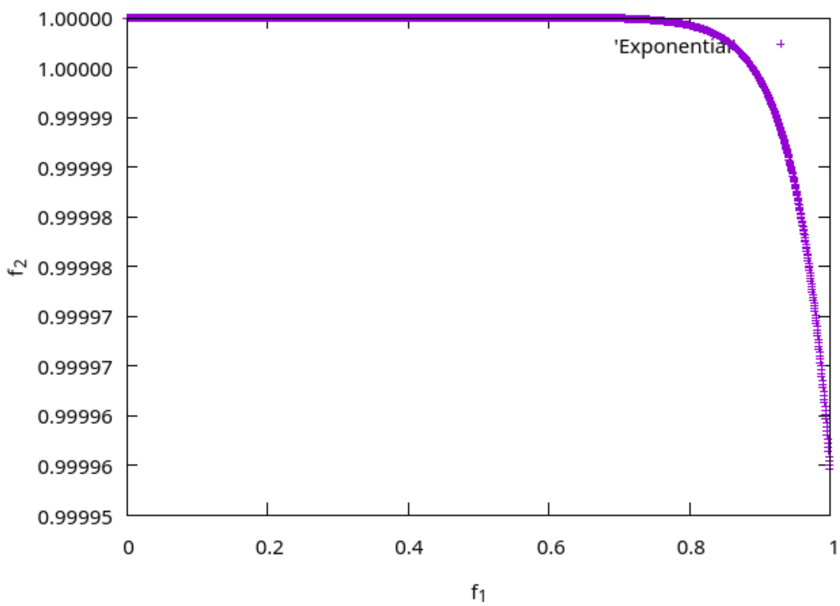

Figure 3

Simplex

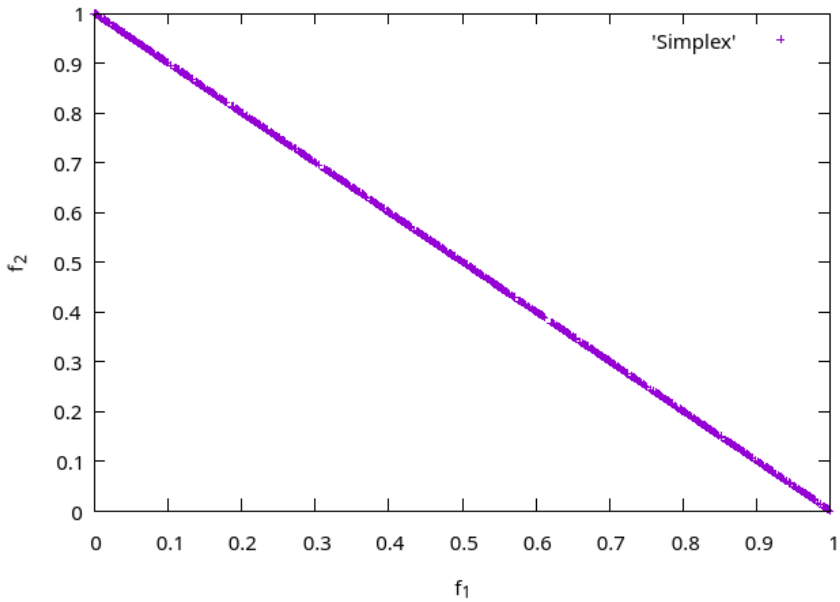

Figure 4

Polynomial

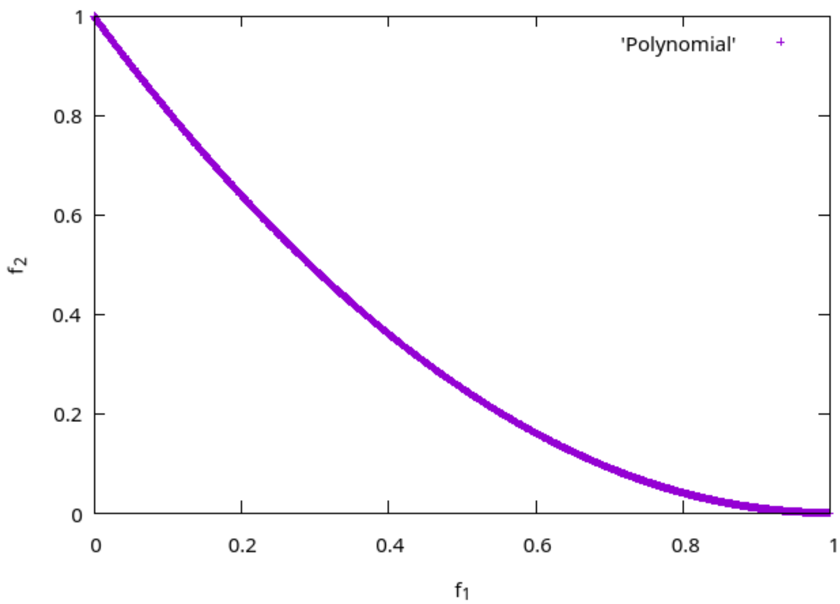

Figure 5

Sqrt

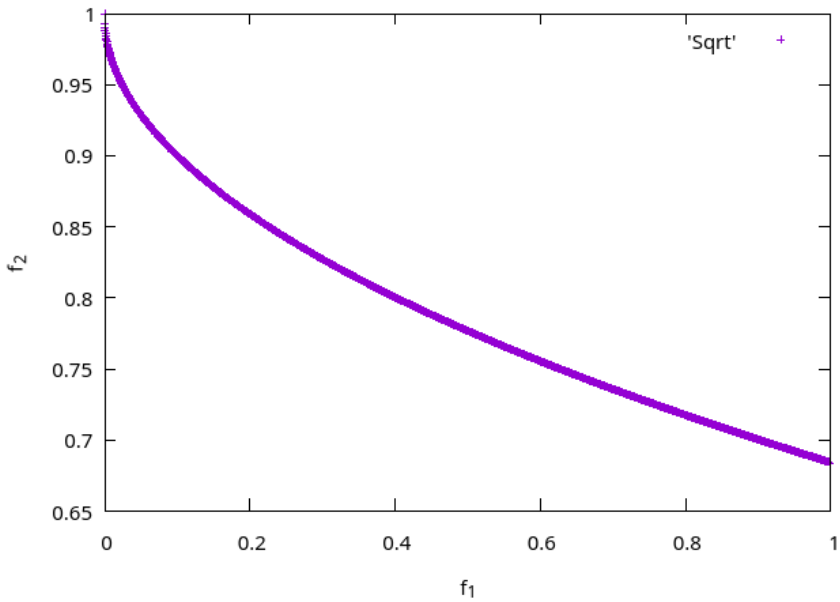

Figure 6

Global optimization approaches

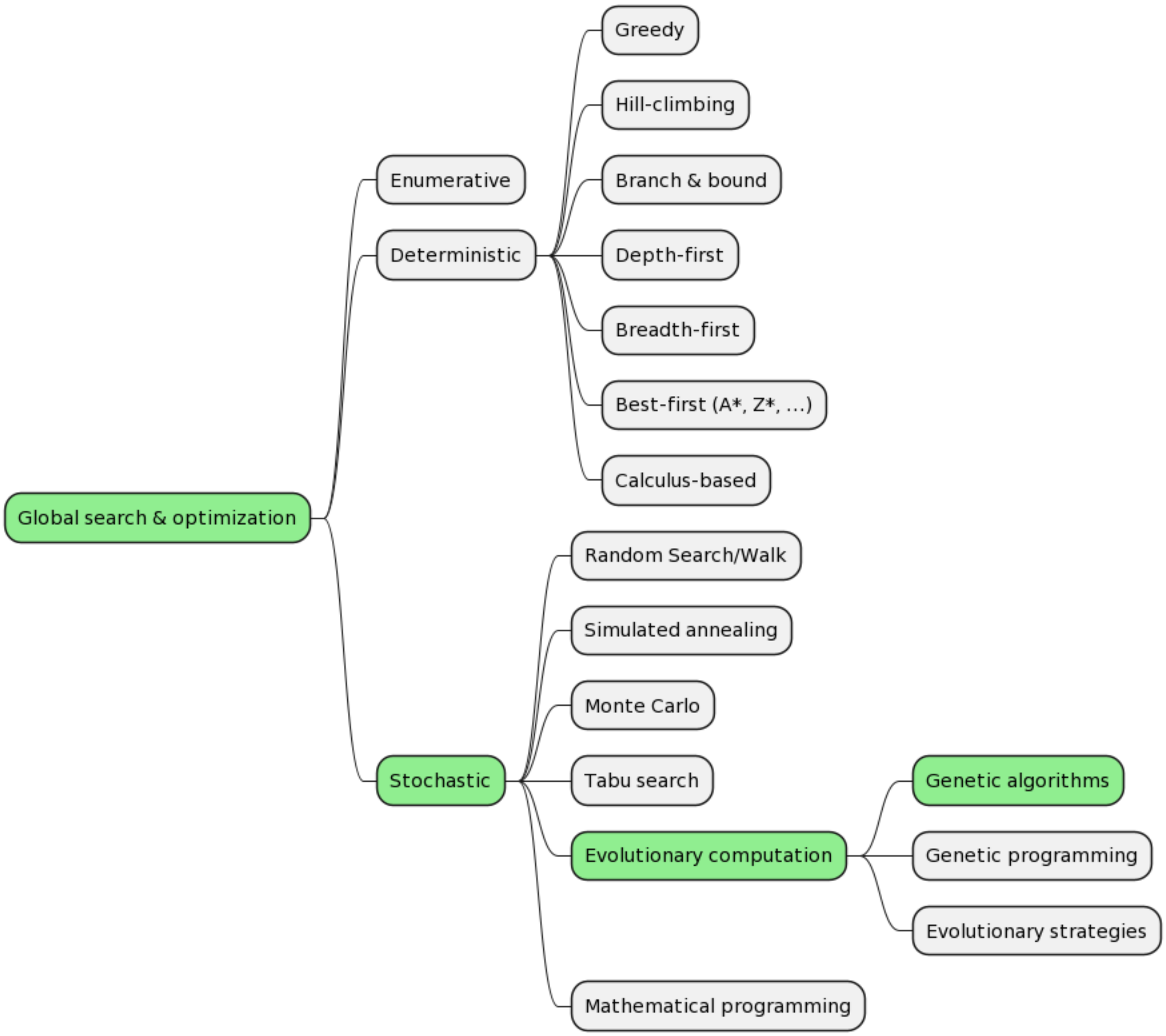

Figure 7

Population

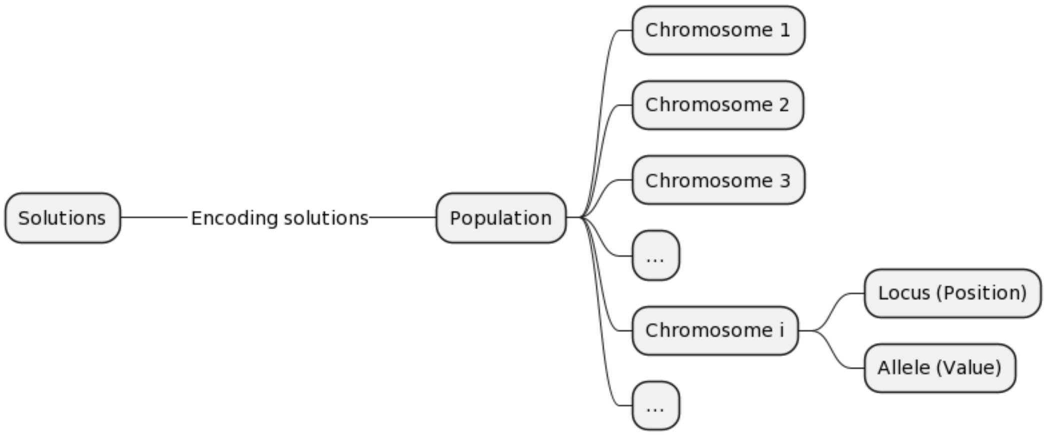

Figure 8

Polynomial

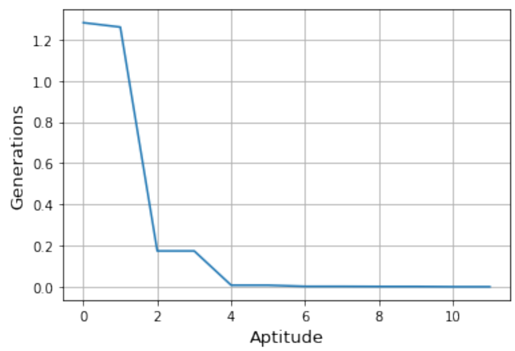

Figure 9

RUP phases and disciplines

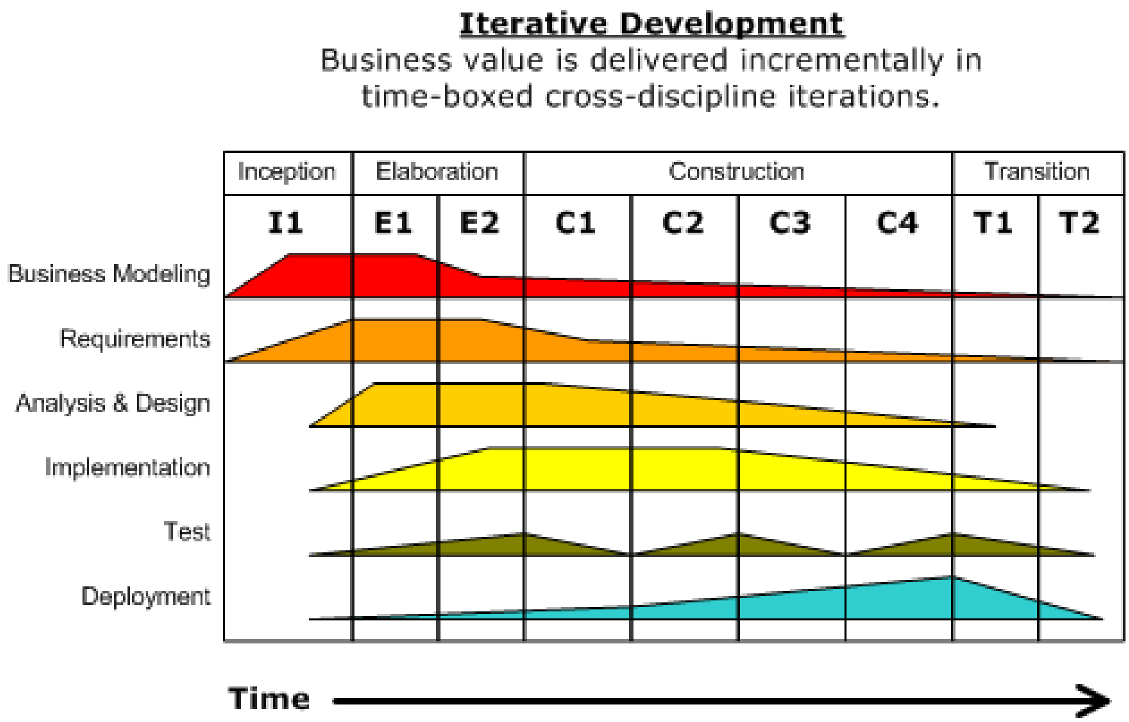

Figure 10

Schedule