

A Parallel Multi-criterion Evolutionary Approach for Pension Fund Asset Liability Management

Christian von Lücken¹ Benjamín Barán¹ Fabián Laufer¹
Margarita Rojas² Mariano Escurra² Walter Delgado²

Universidad Nacional de Asunción¹

Caja Paraguaya de Jubilaciones y Pensiones del Personal de la Itaipú Binacional²
Asunción, Paraguay

Abstract

This paper describes a decision making support system developed for asset liability management of a Paraguayan pension fund company. The model combines a parallel Multi-objective Evolutionary Algorithm and stochastic scenario generators to find good asset allocations that optimize several conflicting criteria. The simple but powerful model has shown to be useful to aid decision maker in practical situations.

Keywords: Artificial Intelligence, Multi-Objective Evolutionary Algorithms, Multi-objective Optimization, Decision making support systems, Asset Liability Management

1. Introduction

A pension fund makes benefit payments to its affiliates who have ended their active income earning career. Facing various uncertainties such as the future life expectancy, inflation or return on investments, the fund manager is supposed to guarantee the benefit payments over a very long period of time.

Asset liability management (ALM) problem concerns with finding an optimal investment policy which determines the pension fund strategic asset allocation in order to guaranty with large probability of success the ability of the fund to fulfill all their liabilities in the long-run under a variety of plausible scenarios [1].

The size of the liability payments and investment returns depends on the state of the world in the future; thus, ALM problems have been usually solved using stochastic programming and simulations [1–4].

An optimal investment policy may not have a single

measure of performance but many, each one measuring different types of returns and risks which may need to be considered at the same time. This work proposes the use of a parallel Multi-objective Evolutionary Algorithm (pMOEA) to aid Decision Makers (DM) in ALM problems considering several conflicting objectives simultaneously. Since evaluating long-term investment strategies requires knowledge on several uncertain economic variables, a scenario generator is used in order to fit past data and trends for the different products that compose an asset class. Considering several possible scenarios, the proposed pMOEA tries to find good policies that optimize various conflicting objectives on the planning scope. Also, this work proposes an interval representation of minimum and maximum percentages of the total investment for a given asset class [5].

This paper is organized as follows: section 2 describes a general asset liability problem, section 3 presents the pMOEA model used in this work. Finally, section 4 presents conclusions and future works.

2. Multi-criterion ALM

This section presents the ALM model and a general multi-objective ALM formulation, using a notation given in [2].

The planning scope is divided in T subperiods of one year each. An index t denotes subperiods, $t = 0$ is the current time and T is the length of the planning horizon. By period $t > 0$, we mean the span of time $[t - 1, t)$.

Investment alternatives of a fund fit in one of n disjoint sets representing major asset categories. The total asset value for category i at the beginning of time t is denoted by X_i^t . Thus, a vector: $\mathbf{x}^t = (X_1^t, \dots, X_n^t)$, represents the current investment distribution by category at

beginning of time t . Value \mathbf{x}^0 is the current state of the fund and thus, it should be known.

The number of investment alternatives for category i is represented by m_i , with $m_i \geq 1$. The j -th investment alternative in i is represented by (i, j) . Then, $X_{i,j}^t$ denotes the total value a pension fund has in (i, j) at the beginning of a given time period t .

Evaluating long-term investment strategies requires knowledge about several uncertain economic variables. To model uncertainty, a set \mathbf{S} of feasible scenarios can be used. Literature describes many scenario generators having capacity to obtain scenario sets [3, 4].

Each scenario represents a possible realization of all uncertain variables in the model, from $t = 1$ to $t = T$. In this work a scenario S is completely defined by:

$$S = \begin{pmatrix} \mathbf{r}^{S,1} & W^{S,1} & L^{S,1} \\ \vdots & \vdots & \vdots \\ \mathbf{r}^{S,T} & W^{S,T} & L^{S,T} \end{pmatrix} \quad (1)$$

where, considering time t and scenario S , the obtained return for investment (i, j) is termed by $r_{i,j}^{S,t}$, and total wages of active participants and total liabilities are termed by $W^{S,t}$ and $L^{S,t}$ respectively. Thus, earnings retained in the fund after the payment of all liabilities at time t and scenario S , denoted by $A^{S,t}$ is:

$$A^{S,t} = \sum_{i=1}^n \sum_{j=1}^{m_i} (1 + r_{i,j}^{S,t}) X_{i,j}^{S,t} + W^{S,t} - L^{S,t} \quad (2)$$

The asset allocation policy of a fund is modeled as a vector of intervals:

$$P = [(l_1; h_1), (l_2; h_2), \dots, (l_n; h_n)]$$

such that:

$$100 \in \left[\sum_{i=0}^n l_i, \sum_{i=0}^n h_i \right]; \quad 0 \leq l_i \leq h_i \leq 100 \quad (3)$$

where l_i and h_i represents lower and upper percentages of total investment allowed for category i .

The policy is determined at $t = 0$ and does not change during the planning scope T . Then, at each time t , the policy takes care of the following restrictions:

$$l_i \leq X_i^{S,t} \leq h_i, \quad \forall i \in \{1, \dots, n\} \quad (4)$$

In order to meet the requirements of the investment policy at the beginning of each period t sell and buy operations may be necessary. Let $\Delta^+ X_{i,j}^{S,t}$ and $\Delta^- X_{i,j}^{S,t}$ represent the amount of (i, j) bought and sold at time t respectively; and let $b_{i,j}^t$ and $c_{i,j}^t$ be proportional costs associated with these operations; then investment value in asset class i at the beginning of period $t + 1$ in S , is:

$$X_i^{S,t+1} = \sum_{j=1}^{m_i} ((1 + r_{i,j}^{S,t}) X_{i,j}^{S,t} - \Delta^- X_{i,j}^{S,t} + \Delta^+ X_{i,j}^{S,t} - c_{i,j}^t (\Delta^- X_{i,j}^{S,t}) - b_{i,j}^t (\Delta^+ X_{i,j}^{S,t})) \quad (5)$$

Considering a given policy, an initial asset distribution and a set of scenarios, it is possible to simulate the

long-term performance of a fund if it uses a given policy P . At each time period a heuristic procedure may obtain a vector of sell and buy values [6]. A number of meaningful performance metrics to evaluate a policy under a set of scenarios can be defined; by example: the mean expected return, return variance, risk of underfunding among others. This way, defining a vector $\mathbf{F}(P) = [f_1(P), \dots, f_k(P)]$ of performance metric functions, it is possible to state a general multi-objective ALM formulation as:

Given: \mathbf{x}^0 , the asset distribution in time 0 and,

$\mathbf{S} = \{S_1, \dots, S_m\}$, a set of feasible scenarios,

find a policy:

$$P = [(l_1, h_1); \dots; (l_n, h_n)]$$

that optimizes:

$$\mathbf{F}(P) = [\mathbf{f}_1(P), \dots, \mathbf{f}_k(P)].$$

3. Multi-criterion ALM optimization using pMOEAs

In general, the solution of a multi-objective optimization problem with conflicting objectives is a set of trade-off solutions that can not be improved in any dimension without degradation in another. Trade-off solutions are called Pareto or non-dominated regarding a given solution set if the set does not contain a better solution when all objectives are taken into account. All non-dominated solutions regarding the whole search space compose the true Pareto set, termed as P_{true} , its image in objective space is named as Pareto front (denoted PF_{true}) [7].

Ideally, decision makers must choose a P_{true} solution. Hence, knowledge about PF_{true} will help DM to choose the best trade-off solution according to his preferences. MOEAs were effectively used to explore huge search spaces in various multi-objective problems and parallel MOEA (pMOEA) have proven to improve MOEA performance [7,8]. Then, this work proposes a combination of pMOEAs and stochastic scenario generators to approximate P_{true} in ALM multi-objective problems. The main elements of the architecture are a scenario generator, an optimization module and a visualization module, as shown in Figure 1.

Financial databases containing historical value of assets and pension fund database of participants provide analytical data. This data feeds a scenario generator which obtains a set of scenarios to be used by the optimization module to evaluate different policies searching for optimal ones [3, 4]. Expert advice is necessary to determine which scenario types should be analyzed.

Population size, the number of generations and the number of evaluated scenarios affect the quality of final solutions. This work proposes a pMOEA based on the multi-deme approach to improve all these factors. In multi-deme evolutionary algorithms, one population

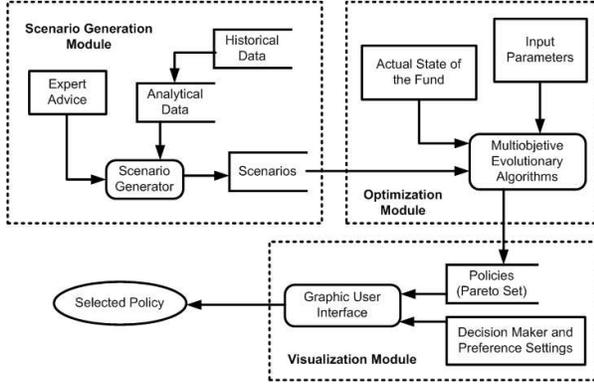


Figure 1: Basic proposed architecture

is divided into subpopulations called islands, regions or demes. Each subpopulation runs a separate evolutionary algorithm. The fitness value of an individual is calculated only relative to other individuals from the same region. Additionally to the basic operators of a genetic algorithm, a migration operator which controls the exchange of individuals between islands is introduced. By dividing the population into regions and by specifying migration criteria, the multi-deme model adapts to various parallel architectures [8].

The proposed parallel model uses a collector and several pMOEAs. The collector (see Pseudocode 1) spawns all pMOEA processes and receives calculated solutions from them. In addition, it maintains an archive of the non-dominated solutions interchanged between demes and provides the final Pareto approximation set. This process does not utilize any evolutionary operator and does not interfere with the evolutionary process done by each pMOEA process. If the number of solutions in the collector exceeds a desired number, a clustering procedure is used to prune the solution set [9].

Meanwhile, pMOEAs perform the real computation. In this work, implemented pMOEAs are based on the Strength Pareto Evolutionary Algorithm II (SPEA-II) [10] as shown in Pseudocode 2. Initially, pMOEAs receive a set of feasible scenarios, the actual state of the fund and specific pMOEA parameters [8]. Then, a randomly generated genetic population (\mathbf{P}) and an empty archive ($\bar{\mathbf{P}}$) are created. Values in \mathbf{P} that do not meet restrictions in Equation (3) are randomly modified until all population elements fit restrictions.

Considering the provided set of scenarios \mathbf{S} , objective values \mathbf{F} are calculated. Performance functions composing vector \mathbf{F} can be selected from a list by the DM before the search procedure begins.

At the end of a time period t it may be necessary to adjust the current distribution of investments by buying or

Algorithm 1 General structure of the collector

Algorithm Collector ()

Read parameters

Spawn H pMOEAs with their corresponding parameters

while $H > 0$ **do**

Wait for pMOEAs solutions

Collect received solutions in collector population P

Eliminate dominated solutions in P

if the number of solutions in P exceeds a given number **then**

Prune S applying the SPEA clustering procedure

end if

if results are marked as finals **then**

$H = H - 1$

end if

end while

Pass final result to visualization module

selling to fit the policy P . In this work, in order to satisfy policy restrictions, a heuristic approach is used to calculate sell and buy values for a given scenario and period t . Once the k optimization objectives are calculated, the SPEA-II fitness assignment procedure is used to rank solutions in \mathbf{P} and $\bar{\mathbf{P}}$. Then, the environmental selection procedure proceeds [10]. If the stop criterion is reached, SPEA-II sends to the collector non-dominated solutions in $\bar{\mathbf{P}}$. If the stop criterion is not satisfied, the algorithm continues applying a mating selection procedure which fills the mating pool using binary tournament selection with replacement on $\bar{\mathbf{P}}$. After the mating pool is filled, crossover and mutation proceed to produce new offsprings in \mathbf{P} . Since, each interval that composes a policy is coded as a pair of floating point numbers, crossover and mutation operators of continuous genetic algorithms are used [11].

Then, as long as a stop criterion is not reached, the evolutionary process continues. As the number of generations G increases, solutions in $\bar{\mathbf{P}}$ are expected to improve. Then, the final solution set obtained by SPEA-II has the potential to be a good approximation of P_{true} .

At each generation, the migration condition is tested. If the migration condition is true, migrants are selected. In this work, the migration condition is based on a probability test. Since there is no unique best solution to migrate, some criterion must be applied. Thus, elements to migrate are considered only among non-dominated solutions in the current archive ($\bar{\mathbf{P}}$). A parameter controlling the maximum number of migrants is provided. Therefore, migration of individuals is controlled by two parameters, one for the frequency of communications, and another for the number of migrants. Migrating elements may represent a fraction of the non-dominated set of individuals that currently are in a MOEA's population [8].

Algorithm 2 General SPEA-II pseudocode

Receive pMOEA parameters, a scenario set \mathbf{S} and \mathbf{x}^0
Set $G = 0$
Generates an initial population \mathbf{P} randomly
Make \mathbf{P} feasible according to constraints
Create an empty archive $\bar{\mathbf{P}}$
Repeat
 Calculate objective vector \mathbf{F} for every solution in \mathbf{P}
 Calculate fitness values of individuals in \mathbf{P} and $\bar{\mathbf{P}}$
 Apply environmental selection (update $\bar{\mathbf{P}}$)
 if the stop criterion is reached **then**
 Send non-dominated solutions in $\bar{\mathbf{P}}$ as finals and exit
 end if
 Apply mating selection
 Use crossover and mutation to obtain a new population \mathbf{P}
 Make \mathbf{P} feasible according to constraints
 if condition to migrate is reached **then**
 Select migrants from $\bar{\mathbf{P}}$ according to a specified criteria
 Send migrants to all other processes
 end if
 if there are received solutions from other demes **then**
 Replace individuals in \mathbf{P} by received ones
 end if
 $G = G + 1$
End repeat

At the end of the optimization procedure, solutions in the collector are passed to the visualization module which presents the solutions to the decision maker. Objective values for each trade-off solution are presented in a visualization grid. This grid let DM perform a lexicographic ordering specifying its preferences and apply filters over some ranges. When a given row in the grid is selected the corresponding policy is depicted using a bar graphic, where each bar represents an interval.

4. Conclusions

This work presents for the first time the use of a parallel Multi-objective Evolutionary Algorithm combined with stochastic scenario generators in order to aid decision making in multi-objective ALM problems. Also, it proposes the use of intervals denoting minimum and maximum percentage of the total investment for a given asset class to represent investment policies [5].

The proposed model have proven to be useful to aid decision making in a real pension fund for which the current work has been developed [12] providing a good number of alternative solutions to choose from.

Due to space restrictions, and the interest of presenting the entire design architecture, experimental results are not included. A further work will cover in details these results and will incorporate decision-maker preferences while the optimization procedure goes on.

References

- [1] W. T. Ziemba and J. Mulvey, eds., *World Wide Asset and Liability Modeling*. Cambridge University Press., 1998.
- [2] S. J. Drijver, W. K. Klein Haneveld, and M. H. v. d. Vlerk, "ALM model for pension funds : numerical results for a prototype model," Tech. Rep. 02A44, University of Groningen, SOM, 2002.
- [3] J. Dupacova and J. Polivka, "Asset-liability management for czech pension funds using stochastic programming," *Stochastic Programming E-Print Series (SPEPS)*, vol. 1, 2004.
- [4] L.-Y. Yu, X.-D. Ji, and S.-Y. Wang, "Stochastic programming models in financial optimization: A survey," *AMO — Advanced Modeling and Optimization*, vol. 5, no. 1, 2003.
- [5] P. W. Oliveira, T. A. Diverio, and D. M. Claudio, *Fundamentos da Matemática Intervalar*. Porto Alegre: Sagra-Luzzatto, 1997.
- [6] B. Barán, C. von Lüken, and A. Sotelo, "Multi-objective pump scheduling optimisation using evolutionary strategies," *Advances in Engineering Software*, vol. 36, pp. 39–47, January 2005.
- [7] C. A. Coello Coello and G. B. Lamont, eds., *Applications of Multi-Objective Evolutionary Algorithms*. Singapore: World Scientific, 2004.
- [8] B. Barán, C. von Lüken, and A. Sotelo, "Pump scheduling optimization using asynchronous parallel Evolutionary Algorithms," *CLEI Electronic Journal*, vol. 7, p. Paper 2, December 2004.
- [9] E. Zitzler and L. Thiele, "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach," *IEEE Trans. on Evolutionary Computation*, vol. 3, pp. 257–271, November 1999.
- [10] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," Tech. Rep. 103, Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology, Zurich, Switzerland, May 2001.
- [11] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. New York: Springer-Verlag, 1994.
- [12] Caja Paraguaya de Jubilaciones y Pensiones del Personal de la Itaipú Binacional, www.cajubi.org.py.