Algorithmic Capital Allocation for Decentralized Applications

Orhan Koc
hankoc@onchainfactory.com

***Abstract*—With the advent of blockchain technology, ownership of purchasing power can be proved without a trusted central entity. This fundamental advancement has made it possible to build semi algorithmic companies, paving the way for a new paradigm. As is the case with multi-division companies, determining the capital allocation strategy of a decentralized application is a crucial component for its growth. This paper provides an implementation of a capital allocation algorithm to compare the alternatives of revenue utilization in a decentralized applications using the principles of risk management and capital allocation.**

I. Introduction

Early age of cryptocurrency projects used to be composed of a financial service which incurs a fee to the user. As the advantages of blockchain based services became more apparent, the design of such applications became more complex, with multiple revenue and expense streams added such as the token, application treasury, node operators et cetera. The newly added corporate structures to simple cryptocurrency applications had a lot of technical debt, in that applications failed to implement common corporate finance strategies to their organization. The seperation of these sub-divisions in trustless systems warrant a clear identification of a comparison framework to evaluate the opportunity cost of alternatives.

It's important to note this paper isn't about capital budgeting, which describes how capital is spent within a sub-division; for example what percentage of the treasury should be allocated for operational expenses, or which venture to acquire in the presence of excess cash. Rather it attempts to automate the decision making process of distributing available capital to different sub-divisions to ensure the strategic plan is met.

A. Literature Review

Capital allocation within portfolios where all assets have a return in terms of dollars is an extensively researched field which are also applied in Capital Allocation within companies.

1) Maximizing expected return: [TODO]

2) Maximizing expected utility: [TODO]

Capital Allocation in decentralized applications is fairly limited due to the relatively recent emergence of advancements in blockchain space. Because of the experimental nature of decentralized applications, budgeting of these financial systems have rather been on the safer side of the spectrum.

3) Revenue compartmentalization: occurs when funds generated by a sub-division are used only within that sub-division. An example would be to buy and burn tokens using token transfer fees alone.

4) Hardcoding: is a simple technique of budgeting by allocating revenue to sub-divisions as predetermined fractions with no updates. Even though it is a fairly foolproof strategy to allocate funds, the lack of adaptability to market conditions render this technique suboptimal. This approach is parallel to history-based capital allocation where ratio of funds allocated are equal to that of previous year.

5) DAO Voting: is a democratic fund allocation process where every token counts as one vote. The idea is more stake in the system should equal more authority in making decisions. This approach is very similiar to Participatory Budgeting which is a paradigm that empowers residents to directly decide how a portion of the public budget is spent [1].

B. Proposal

The fundamentals layed out in Capital Asset Pricing Model (CAPM) [2] have been useful in modeling risk and return of assets bearing risk. Even though the risk and return of sub-divisions of a company differs from that of an asset's, there have been successful attempts at modeling budgeting algorithms using portfolio management theories [3]. We propose a standardized, data-driven algorithm to decide the allocation for sub-divisions of a modern cryptocurrency application.

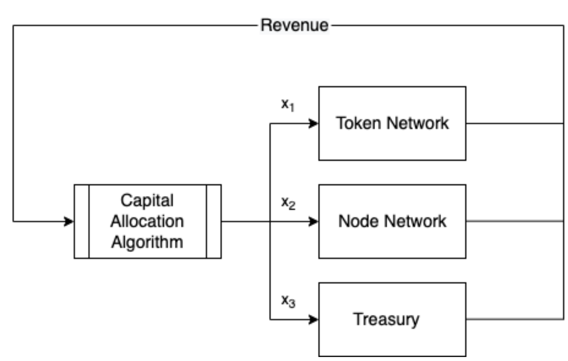

Fig. 1 High-Level Revenue Flow

The success of capital allocation strategy is determined by whether Key Performance Indicators as proposed by the executive board (KPI) are met or not. We propose to formulate a capital allocation algorithm in the form of a stochastic optimization problem for n sub-division and record the incremental change in KPI per dollar spent (u_n) with respect to funds allocated (x_n).

In order to calculate the incremental change in KPI, (u_n), the algorithm calculates

The algorithm is optimized for the highest expected

The algorithm allocates funds to meet the KPI targets for the remaining period of time according to corresponding fund utilization feedback.

We will base our model to be applicable for an application with 3 alternatives: Nodes, Token Holders and Treasury.

We will then specify constraints for this optimization problem based on **Capital Budgeting Risk Budgeting** to specify the solution space. In the end, we will have a model with decision variables as funds allocated (x_n) to subdivision n and implement an iterative process to solve for the optimal value of capital allocation that maximizes the expected return on KPIs.

II. Theory

We first acknowledge that we are trying to optimize for the allocation of limited resources to a set of options that results in unexpected returns. The most common way to approach a budgeting problem is to use Linear optimization to maximize a The general form for linear optimization model is as follows:

Minimize or maximize:

$$Z = \sum_{i=1}^{n} c_i x_i \tag{1}$$

Subject to:

$$\sum_{i=1}^{n} a_i x_i \leq b_i; \tag{2}$$

Where Z is the objective function subject to minimization or maximization, $c_1, c_2, ..., c_n$ are the coefficients of the corresponding decision variables, $x_1, x_2, ..., x_n$ are the decision variables we are solving for, $a_1, a_2, ..., a_n$ are the constraint coefficients and $b_1, b_2, ..., b_n$ are the constraints.

III. Model

A. Basic setting

The challenge of formulating a capital allocation optimizaton problem for a decentralized application is to find a single objective function that is applicable for different subcomponents. The simplest example is that the performance of a Treasury is measured in terms of Return on Investment, but that of a node network is measured in terms of staking percentage.

The variables(x_i) are simply the expenditure on a subcomponent. The variable coefficients(c_i) should represent the weight of the subcomponents' expenditure to the objective. The constraints can be modified to fit the needs of specific cases. The constraints, denoted in Equation 2, are the constraints that must be satisfied in order for a solution to be considered feasible in the linear optimization problem.

B. Design

The main goal of using such an algorithm would be to reach the *strategic plan* determined by the executive board. In order for the strategic plan to be applicable to the aforementioned optimization model, the plan should include performance targets for the Treasury, Node and Token.

1) Variable Coefficients: Essentially, we want to maximize the overall target KPI completion using a fixed amount of funds; to model this objective, we need a complementary coefficient that will convey the relationship between subcomponent expenditure and performance increment, α_i.

-	*KPI (P)*	Δ *KPI*
Treasury	Growth	$\frac{R_t - E_t}{C_{t-1}}$
Token	Inflation	$\frac{M_t - B_t}{T_{t-1}}$
Node	Staked amount	$\frac{L_t - U_t}{S_{t-1}}$

For treasury, R_t is revenue of the protocol added to treasury, E_t is the expenses that are deducted from treasury and C_{t-1} is the treasury size at the previous observation so that $\Delta\ C_t$ is the numerator.
For token, M_t is the amount of newly minted tokens, B_t is the amount of newly burned tokens and T_{t-1} is the total amount of tokens in circulation at the previous observation so that $\Delta\ T_t$ is the numerator.
For node, L_t is the amount of newly staked tokens, U_t is the amount of newly unstaked tokens and C_{t-1} is the total amount of staked tokens at the previous observation so that $\Delta\ S_t$ is the numerator.

Our objective is to maximize the total rate of improvement with respect to performance targets.

$$c_i = \frac{\alpha_i}{P_t - P_i}$$

c_i represents expected incremental improvement in KPI per dollar in expenditure. P_t P_c

2) Objective:

$$Z = \frac{\alpha_i}{P_t - P_i} x_i$$

It's important to realize KPIs chosen should be directly affected by the subcomponent expenditure(x_n).

For example the funds spent on token inflation, utilized by buying the tokens from the market and burning them, have a direct effect on the inflation: the number of tokens put into circulation divided by number of tokens removed from circulation. Instead, if we are to use token price as a KPI for tracking funds spent on the token network the outcome would be noisy, as the price is affected by many other parameters.

3) Constraints: From a capital allocation perspective, probabilistic constraints of the model correspond to the limits on how much fund can be spent on each subnetwork.

The most common seperation of these limits as seen in portfolio optimization models are the **Risk Constraint** and **Budget Constraint**, which can also be applied in this context. **Risk Constraint** defines the maximum **Budget Constraint**

C. Solution Method

IV. Implementation

For the sake of simplicity, we will consider the simplest cryptocurrency application with three essential divisons: token network, node network and the treasury.

V. Conclusion

A. Results

B. Evaluation

C. Limitations

References

[1] H. Aziz and N. Shah, "Participatory budgeting: Models and approaches," in *Pathways Between Social Science and Computational Social Science*, pp. 215–236, Springer, 2021.

[2] H. Markowitz, "Portfolio selection," *The Journal of Finance*, vol. 12, pp. 77–91, 1952.

[3] A. R. Cetingoz, J.-D. Fermanian, and O. Guéant, "Stochastic algorithms for advanced risk budgeting," *arXiv preprint arXiv:2211.07212*, 2022.