

論文内容の要旨

論文題目

Development of Hybrid Evolutionary Techniques to Solve
Resource Allocation Problems
(資源配分最適化問題への進化論的アプローチ)

氏名 アランニャ クラウス デ カストロ

The Resource Allocation problem is any problem that can be defined as the division of a "Resource" among a large number of "Consumers". The "Consumers", in this case, are parameters of an function optimization problem. Two examples of Real World Resource Allocation problems are the Financial Portfolio Optimization, and the Economic Load Dispatch problems.

In the Financial Portfolio Optimization the "Resource" is the Financial Capital, and the "Consumers" are the different assets and securities that a trader can invest using this capital. A trader wants to form a Portfolio with many different assets in order to reduce the risk, i.e., the chance that the value of the portfolio will change suddenly. A well chosen portfolio will have a very low risk with a reasonable return, and is important for long term financial investments such as retirement funds, insurances, etc. Thus, the Financial Portfolio Optimization problem has great relevance in the Financial Engineering Field.

In the Economic Load Dispatch problem, the "Resource" is the total energy demand of a Power Supply System, and the "Consumers" are the Energy Generators in this system. Each Generator has a minimum and a maximum output, and different levels of fuel consumption for different output levels. By distributing the energy demand correctly among the generators, the whole system will not only consume less fuel, but also work more reliably and with a smaller environmental impact.

In this work, we develop a hybrid optimization system specialized in solving Resource Allocation problems, called the MEMETIC TREE-BASED GENETIC ALGORITHM (MTGA). The MTGA is based on evolutionary algorithms (EA), and uses some novel

improvements to the EA framework. EAs are population-based optimization heuristics inspired in the natural evolutionary process. These Evolutionary Algorithms have been shown to be successful in hard and high dimensional problems. We develop three new techniques that together greatly improve the performance of EAs in the Resource Allocation domain: a specialized problem representation, a local search operator, and a topological global search framework.

The first new technique is the specialized problem representation. The traditional way to represent a problem's solution in an evolutionary algorithm (referred to as a "Genome") is to use an array where each element in the array is one parameter of the solution. However, in many Resource allocation problems, specially those with a high degree of epistasis, it is also very important to know how different parameters are related to each other. Our new representation, called the tree-based representation, is composed by a binary tree, where the terminal nodes are the parameters of the problem, and the intermediate nodes are local weights that determine how the resource is divided among the parameters. Using this representation, it is possible to create a hierarchical relationship tree between the parameters, which gives us parameter relationship information, which can be used to solve a given problem more efficiently.

The second new technique is the local search optimization operator. It is well known that in real valued problem domains, the use of a local search optimization operator greatly improves the performance of Evolutionary Algorithms. This hybridization is usually called "Memetic Algorithms" (MA). However, the implementation of the search optimization operator is not always straightforward, depending on the problem domain. On highly dimensional problems, trying to locally optimize all the dimensions at the same time may be too costly. We have developed a local search optimization which uses the tree-structure described previously to effect a divide-and-conquer policy which allows the operator to locally optimize small parts of the solution at a time, in a much simpler and faster fashion. In our local search operator, a tree representation is divided into sub trees, and each sub-tree, from the bottom to the top, is optimized individually before being re-inserted into the original solution. In this way, we avoid trying to solve a high-dimensional optimization problem and replace it with many low-dimensional problems.

The third and last new technique studied in this work to for the MTGA optimization system is the Topological Framework. In very hard optimization problems, even Evolutionary Algorithm can display sub-optimal results. This happens mainly when a problem has too many deceptive local optima, and the algorithm can not consistently find the global optima. A way to avoid having an EA getting stuck into a local optima is to promote its "Diversification". Diversification means different procedures to guarantee that the solutions in the population explore a wide area in the search space. The diversification procedure we study here is the Topological Strategy (TS). In the TS we distribute the solutions of a EA in a grid, and we only allow recombination between solutions which are in the same cell in the grid. At

every iteration of the algorithm, each solution has a chance of moving to a neighbor cell. This movement creates a dynamic where the best solutions form "islands", which aggregate other solutions around them. Each island is then able to explore a different area of the search space, reducing the chance that the system as a whole will be trapped in a local optima.

In order to analyze the above MTGA system, we perform three sets of experiments.

The first set is in the Portfolio Optimization problem Domain. We execute a simulation using historical price data from the NASDAQ and the S&P 500 datasets, in order to evaluate the portfolios generated by the MTGA and compare them with a state of art evolutionary optimization system (differential evolution) and a traditional financial benchmark. We simulate 72 different scenarios between 2006 and 2008. The results in these experiments tells us that the MTGA is able to outperform the other comparative techniques, and has proved itself to be an effective system for the optimization of financial portfolios. A deeper analysis of the results show that the MTGA is able to both select the best assets to compose the portfolio and to optimize their relative weights at the same time, while previous methods are only able to perform one or the other. Also, the tree representation introduced in this work was able to learn the structure of the problems, and the best individuals generated were those which featured structures that included problem domain knowledge.

The second set of experiments involved mathematical functions which are often used to benchmark parameter optimization problems. Our objective was to access whether the MTGA has the ability to solve problems other than the financial Portfolio Optimization problem. Our tests included problems with 20 and 200 dimensions, of varying difficulty. In the results, the MTGA showed a very high degree of reliability with regards to the dimensionality of the problem. While in the cases with lower dimensions all the methods performed comparably, the MTGA showed a reduced loss in performance for higher dimensions when compared to the other methods.

In the third set of experiments, we apply the MTGA to a set of problems from the Economic Load Dispatch domain. There problems include benchmarks from some seminal works in this problem. The results in this experiment highlight the the abilities and limitations of the MTGA system, and allow us to outline the problem domains in which the MTGA is able to excel.

In the end, we developed a system which shows a significant superior performance in the Portfolio Optimization domain by applying the principle of including domain knowledge to the problem representation. This principle shows that the evolution paradigm for optimization problems is much stronger than previously though, and is able to learn extra information about a problem domain, as long as the representation allows for such information to be stored in it. Our system provides a distinct service in this real world application, and indicates the ability to be useful in other resource allocation problems.