MOAP is a novel approach to automated program synthesis, focusing on automated algorithm design. Algorithms are the key steps, or recipes, for a program. We introduce a new language for representing algorithm problems, consisting of a "given" and a "goal".

Our approach is centered around a rule-based reduction of these problems. There are a series of algorithm "chunks", each of which removes a part of the algorithm problem when used. The program performs a "great flood" search from the initial problem, trying all possible applicable algorithm stubs at each problem state, until it reaches a problem state where the goal is reduced to nothing, at which point it has reached a solution. The resulting algorithm is generated by backtracking through the algorithm stubs used to find the solution.  
  
To guarantee an optimal solution and minimize branching time, the program weighs branching options by the projected runtime of each algorithm stub chain. By branching the best algorithm chains first, the program guarantees that the solution reached will be the best (by whatever criteria), as well as avoiding superfluously structured algorithms.

Currently, MOAP is still in its infancy, but is able to solve a fair range of algorithm problems using a strong set of algorithm stubs. The brunt of the work of the project is related to the expansion and refinement of the algorithm stub database.

We hope that this approach will serve to transform the world of computer science, reducing algorithm design to something as automated as mathematics is today.