7.1 Combinatorial Optimization Problems

Ming Lu

2022-04-02

Definition 7.1 An optimization problem is a tuple , where is a set of feasible solutions, is a set of constraints over a fixed number of parameters, and is an objective function to be minimized. The set of constraints consists of all the variables in a given set of equations. The set of feasible solutions consists of all arrays of values for the variables that satisfies the constraints in . A solution for the optimization problem is an element of such that for all ; such a is called a globally optimal solution.

In a strict sense we have defined above a minimization problem. For a maximization problem just ask for the global solution to satisfy for all .

tormally treat any optımızation problem as a mınımızation problem. Example Consider a GARCH(1,1) model for a log return series ,

with . Here we want to estimate the parameters and , so that we can recursively obtain and . We have seen in Chap. 4 that this estimation can be done by the method of maximum likelihood. Thus, the underlying combinatorial optimization problem can be explicitly define as follows. The set of constraints consist of Eq. (7.1) plus the inequalities involving tuples of four variables ; the set of feasible solutions are the values for that satisfy the Eq. (7.1) and inequalities, and we seek for those values that maximizes the log likelihood function (without additive constants)

This is the objective function to consider for this problem, where one seeks to maximize its value with respect to all . This function can have various local maxima, and therefore a traditional iterative numerical method (e.g. the local search described below as Algorithm 7.1) will often fail to find a global optimum. Hence, the need of an optimization heuristic is justified.

Example 7.2 Another maximization problem we consider in this chapter is to find trading rules from Technical Analysis that best perform in a period of time and for an asset’s return time series . A description of this problem as a tuple is given by: as a set of production rules which describe the process of constructing technical trading rules (or investment recipes as those presented in Sect. 6.1), from some predefined set of constants, variables and functions. consists of the trading rules conforming to the specifications given by . could be the compounded benefit of a trading rule acting on (for an absolute measure of performance), or the excess return over the whole period with respect to a buy-and-hold strategy, which is a more realistic measure of performance as it is a test against the most common passive investment attitude. The objective is to find the trading rule in with maximum -value when acting on . Note that the search space of all possible trading rules is exponential in the number of basic elements used for defining trading rules; hence an exhaustive deterministic search is out of the question, and we can justly turn to heuristic optimization techniques.

The classical local search algorithm for solving an optimization problem starts with an initial solution (usually generated at random), and by making small perturbations or changes to the current solution creates another; that is, new solutions are obtained “in the neighborhood of current solutions”. If the new solution is better (i.e. has a smaller -value) then the current solution is substituted by the new one. This process is repeated until some stopping criteria is met, which can be (and often is) that there is no further significant improvement in the values of , or that a predetermined number of repetitions have been accomplished. Algorithm 7.1 presents the classical local search procedure for solving . Observe that with this procedure we can only guarantee that the search ends in a local optimum. In other words, if our optimization problem has more than one local minimum, once the search routine reaches one of these local solutions, it is likely to get trap in its vicinity because small perturbations will not produce better solutions, and so it misses the global minimum. A way to escape from this local trap is to allow to enter into the search path some solutions that are worse than those previously obtained. However, these “escape-moves” (also known as uphill-moves) must be performed in a controlled manner to avoid jumping out of the vicinity of a potential global solution. The heuristics presented in the following sections are in essence different implementations of these escape-moves.