Branch and Bound

Consider the following problem: find $x_1, x_2, \ldots, x_{10} \in \mathbb{Z}$ to maximize

$104x_1 + 128x_2 + 135x_3 + 139x_4 + 150x_5 + 153x_6 + 162x_7 + 168x_8 + 195x_9 + 198x_{10}$

such that

$9x_1^2 + 8x_2^2 + 7x_3^2 + 7x_4^2 + 6x_5^2 + 6x_6^2 + 5x_7^2 + 2x_8^2 + x_9^2 + x^2_{10} \leq 6864$

A Branch and Bound Algorithm attempts to solve this kind of optimization problem by traversing the tree of possible variable assignments, using the following observations:

- At any time, suppose we have a partial assignment of variables, and an upper and lower bound on the valuethe maximum attains.
- If we relax the remaining variables to be real valued, and solve for the maximum, if the result falls below the lower bound *there is no way to complete the variable assignment with integers that will surpass the lower bound.*
- In this case, we know immediately that the partial assignment must be incorrect, and we can backtrack.
- If we complete a variable assignment, and the result gives a value above the lower bound, then this variable assignment provides a *better* lower bound for the full problem, and we can utilize it moving forward.

Applying these ideas recursively, we can test assignments for some variables, determine whether they are feasible, and either explore deeper, or backtrack and test other values for variable assignment. This lets us prune the spaceof possible variable assignments down, and efficiently identify a maximizing assignment.