

A CONCEPTUAL MODEL FOR OPTIMIZATION PROBLEMS

JEAN P. MARTINS

ABSTRACT.

Definition 0.1 (Component). *In a combinatorial optimization problem, a component $c \in C$ is the elementary unity of a solution.*

Definition 0.2 (Search space). *The search space S of a combinatorial optimization problem is a subset of the power set of components $S_{\leq n}(2^C)$ (only S from now on), in which elements $x \in S$ can be composed of a maximum of n components. In practical terms, S is implicitly defined by C and n .*

Example 0.1 (Knapsack problem component). *In a knapsack problem, a component is an item. There are n available items (components) $C = \{1, 2, \dots, n\}$. Therefore, elements $x \in S$ can be composed of a maximum of n items, i.e. $|C| = n$.*

Example 0.2 (Minimum spanning tree component). *In a minimum spanning tree problem, a component is an edge. Considering a undirected complete graph $G = (V, E)$, with $|V| = n$, there are $|C| = (n^2 + 1)/2$ available edges (components).*

Definition 0.3 (Neighborhood). *A neighborhood function $N_k(x) = \{y \in S : d(x, y) = k\}$ defines the neighbors of x in terms of components, with $k \geq 0$ denoting the number of components in which x and y differ. The size of $N(x)$ is usually $|C| \in O(n^k)$.*

Example 0.3 (Add/remove neighborhood). *Solutions $x, y \in S$ are neighbors if they differ in only one component, i.e. $k = 1$. A solution y can be generated from x by adding or removing a component.*

Example 0.4 (Swap neighborhood). *Solutions $x, y \in S$ are neighbors if they differ in two components, i.e. $k = 2$. A solution y can be generated from x by removing a component and adding another component.*

Example 0.5 (2-opt neighborhood). *Solutions $x, y \in S$ are neighbors if they differ in two components, i.e. $k = 2$. A solution y can be generated from x by removing two components (edges) (a, b) and (c, d) and adding the components (edges) (a, d) and (c, b) .*

Observe that the neighborhoods add/remove and swap are independent of the component structure, whereas 2-opt assume the components are edges in a graph.