# An unified model for optimization problems

Jean P. Martins
jeanp@dei.uc.pt

University of Coimbra, Portugal

## 1 Global Optimization

Global optimization concerns finding the best possible solution for *optimization problems*.

### 1.1 Optimization Problems

**Definition 1 (Optimization problem).** *An optimization problem is a tuple* $(U, P, f, \mathcal{I})$*, where* $U$ *is a solution space,* $P : U \mapsto \{true, false\}$ *is a feasibility predicate,* $f : U \mapsto \mathbb{R}$ *is a cost function and* $\mathcal{I}$ *is a set of instances.*

**Definition 2 (Feasible solution).** *Given an instance* $I \in \mathcal{I}$*, a solution* $u \in U$ *is said feasible if it satisfies* $P_I(u)$*.*

**Definition 3 (Feasible set).** *Given an instance* $I \in \mathcal{I}$*, the set* $S_I = \{s \in U : P_I(s)\}$ *is called the feasible set of* $I$*.*

An *optimization problem* is a abstract concept, in the sense that many parameters needed to define it are absent. *Instances* make an optimization problem concrete by providing values for such parameters (Stützle, 1999).

All the instances $I \in \mathcal{I}$ from a optimization problem share the same solution space $U$. However, the feasibility predicate $P_I$ and the cost function $f_I$ are instance-dependent and may induce different *feasible sets*. A solution for an instance of an optimization problem is:

$$s^\star := \arg\min_{s \in S_I} f_I(s), \tag{1}$$

and $s^\star$ is called a global optimum solution.

### 1.2 Combinatorial Optimization Problems (Michiels et al., 2010)

A combinatorial optimization problem (COP) is an optimization problem in which the solution space $U = 2^C$ is the power set of a finite number of components $C$. Each solution $u \in U$ is a subset $u \subseteq C$, and $C$ is called the ground set of $U$ (or *component set*).

The cost of a feasible solution $s \in S_I$ it is a function of the cost of its components $w_I(c) > 0$, $\forall c \in s$. In the case of linear COPs, an optimal solution for an instance $I \in \mathcal{I}$ is:

$$s^\star := \arg\min_{s \in S_I} \sum_{c \in s} w_I(c). \tag{2}$$

The combinatorial nature of a COP is captured by the feasibility predicate $P_I$, which limits the components that can co-occur in feasible solutions.

## 2 Local Optimization

Local optimization is a more restricted way to describe optimization problems, in which the solution space $U$ supports a distance metric $d : U \times U \mapsto \mathbb{R}$. In such circumstances a notion of neighboring solutions becomes available.

**Definition 4 ($k$-neighborhood).** *Given a solution space $U$, a distance metric $d : U \times U \mapsto \mathbb{R}$ and a radius $k > 0$, a solution $s \in S$ has neighborhood $\mathcal{N}_k(s) := \{v \in S : d(s, v) \leq k\}$.*

Observe that the neighboring solutions depend on $d$, therefore, different distance metrics induce different neighborhoods.

**Definition 5 (Local optimization problem).** *A local optimization problem is an optimization problem in which the feasible set is embedded with a neighborhood structure $\mathcal{N}_k : S \mapsto 2^S$.*

Since $d$ is defined over the whole solution space $U$, neighborhoods are problem-dependent, but instance-independent. In this context, a solution for an instance of a local optimization problem is:

$$\hat{s} := \underset{s \in \mathcal{N}_k(\hat{s})}{\arg\min} f_I(s), \tag{3}$$

and $\hat{s}$ is called a local optimum solution according to $\mathcal{N}_k(\hat{s})$. If $\hat{s}$ it always guaranteed to be also a global optimum, $\mathcal{N}_k$ is called an *exact neighborhood*.

The search for a local optimum solution can be seen as walk on the *neighborhood graph*, which stops when no improving solutions can be found (Stützle, 1999).

**Definition 6 (Neighborhood graph).** *Given a symmetric neighborhood structure $\mathcal{N}$, in which $s \in \mathcal{N}(v) \iff v \in \mathcal{N}(s)$, the neighborhood graph $G = (S, E)$ is composed of vertices $s \in S$ and edges $(s, v)$ $\forall v \in \mathcal{N}(s)$.*

Methods which explore the *neighborhood graph* in order to find *locally optimum* solutions are referred as *local search methods*.

## 3 Constructive Heuristics

## 4 Combinatorial Optimization Problems

In combinatorial optimization the extra information about the *component set* can be used to define an elementary distance metric over the solution space $U$.

**Definition 7 (Elementary distance).** *An elementary distance metric $d_e : U \times U :\mapsto \mathbb{R}$ is defined in terms of the components shared, or not, by both solutions.*

$$d_e(u, v) := |u - v| + |v - u|, \ \forall u, v \in U \tag{4}$$

The elementary distance metric $d_e$ might be used to induce two non-symmetrical elementary neighborhoods in $U$.

**Definition 8 (ADD-neighborhood).** *Given a solution space $U$ and a distance metric $d_e$, a solution $u \in U$ has neighborhood $\mathcal{N}_{\text{ADD}}(u) := \{v \in U : d_e(u, v) = 1 \wedge u \subset v\}$.*

**Definition 9 (SUB-neighborhood).** *Given a solution space $U$ and a distance metric $d_e$, a solution $u \in U$ has neighborhood $\mathcal{N}_{\text{SUB}}(u) := \{v \in U : d_e(u, v) = 1 \wedge v \subset u\}$.*

The composition of ADD and SUB-neighborhoods produces an elementary neighborhood structure over $U$ which is symmetrical.

**Definition 10 (Elementary neighborhood).** *Given a solution space $U$ and a distance metric $d_e$, a solution $u \in U$ has elementary neighborhood $\mathcal{N}_e(u) := \{v \in U : v \in \mathcal{N}_{\text{ADD}}(u) \bigcup \mathcal{N}_{\text{SUB}}(u)\}$.*

Elementary neighborhoods are problem-independent and are called elementary because more complex and problem-specific neighborhoods can be defined from them.

## 5 Augmented Solution Spaces

**Definition 11 (Partial solution).** *A solution $s \in S$ is said partial if $\exists i \in \mathcal{I} : x_i^s = \star$, and we say $c_i$ is in an undefined state.*

**Definition 12 (Complete solution).** *A solution $s \in S$ is said complete if $\nexists i \in \mathcal{I} : x_i^s = \star$. Every component $i \in \mathcal{I}$ is present ($x_i = 1$) or absent ($x_i = 0$) states.*

**Definition 13.** *A linear combinatorial optimization problem $(S, w)$, consists in finding an $s \in S$ with minimal (maximal) objective value $f(I(x))$.*

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

*Example 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 $|C| = (n^2 - n)/2$ available edges (components).

**Definition 14 (Neighborhood).** *A neighborhood function $N_k(x) = \{y \in S : d(x, y) = \delta(k)\}$ 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 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 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 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.

## TODO

"Construtive algorithms generate solutions from scratch by adding components to an initially empty solution until completion. They are typically the fastest approximate methods, yet they often return solutions of inferior quality when compared to local search algorithms" (Stützle, 1999).

- Define the current modeling
- Justify the need for the unified model
- Define the extensions
- Model well-known methods and neighborhoods using the extensions

# Bibliography

W. Michiels, E. Aarts, and J. Korst. *Theoretical Aspects of Local Search*. Springer Publishing Company, Incorporated, 1st edition, 2010. ISBN 3642071481, 9783642071485.

T. G. Stützle. *Local search algorithms for combinatorial problems: analysis, improvements, and new applications*, volume 220. Infix Sankt Augustin, Germany, 1999.