Computational Intelligence
Definition Sheet

Thomas Gabor
LMU Munich

Maximilian Zorn
LMU Munich

Claudia Linnhoff-Popien
LMU Munich

winter term 2024/2025

Notation. $\mathbb{B}$ is the set of truth values or Booleans, i.e., $\mathbb{B} = \{0, 1\}$. $\mathbb{N}$ is the set of natural numbers starting from zero, i.e., $\mathbb{N} = \{0, 1, 2, ...\}$ so that it holds that $\mathbb{B} \subset \mathbb{N} \subset \mathbb{Z} \subset \mathbb{R} \subset \mathbb{C}$. $\mathbb{P}$ denotes the space of probabilities and $\mathbb{F}$ denotes the space of fuzzy values with $\mathbb{P} = \mathbb{F} = [0; 1] \subset \mathbb{R}$ being only discerned for semantic but not mathematical reasons. $\mathbb{E}$ denotes the expected value.
$\wp(X) = 2^X$ denotes the power set of X. $+\!\!+$ denotes vector or sequence concatenation, i.e., given two vectors $\mathbf{x} = \langle x_1, ..., x_{|\mathbf{x}|}\rangle$ and $\mathbf{y} = \langle y_1, ..., y_{|\mathbf{y}|}\rangle$, $\mathbf{x} +\!\!+ \mathbf{y} = \langle x_1, ..., x_{|\mathbf{x}|}, y_1, ..., y_{|\mathbf{y}|}\rangle$. A vector $\langle x_0, ..., x_{n-1}\rangle$ with length $n \in \mathbb{N}$ can also be written as $\langle x_i\rangle_{0 \leq i \leq n-1}$ for a new iteration variable i. A sequence containing an arbitrary amount of elements from the space X has the type $\langle X\rangle$.
$\_$ denotes unspecified function arguments ($f(\_) = 0$ is the constant function that always returns zero, e.g.). For any finite set $X = \{x_0, ..., x_n\}$, $|X| = n$ denotes the number of elements in X. For infinite sets, $|X| = \infty$.

Definition 1 (agent)**.** Let $\mathcal{A}$ be a set of actions. Let $\mathcal{O}$ be a set of observations. An agent A can be given via a policy function $\pi : \mathcal{O} \to \mathcal{A}$. Given a time series of observations $\langle o_t\rangle_{t \in \mathcal{Z}}$ for some time space $\mathcal{Z}$ the agent can thus generate a time series of actions $\langle a_t\rangle_{t \in \mathcal{Z}}$ by applying $a_t = \pi(o_t)$.

Algorithm 1 (brute force (policy))**.** Let $\mathcal{A}$ be a set of actions. Let $\mathcal{O}$ be a set of observations. Let $\Gamma \subseteq (\mathcal{O} \to \mathcal{A}) \to \mathbb{B}$ be a space of goal predicates on policy functions. Let $\gamma \in \Gamma$ be a goal predicate. We assume that the policy space $\Pi \subseteq \mathcal{O} \to \mathcal{A}$ is enumerable, i.e., $\Pi = \langle \pi_i \rangle_{i \in \mathbb{N}}$. Brute force starting from i is given via the function

$$b(i) = \begin{cases} \pi_i & \text{if } \gamma(\pi_i), \\ b(i+1) & \text{otherwise.} \end{cases}$$

If not further specified, the call to $b(0)$ is called brute force search for an agent policy. Usually, an additional termination condition is specified.

Algorithm 2 (random search (policy))**.** Let $\mathcal{A}$ be a set of actions. Let $\mathcal{O}$ be a set of observations. Let $\Gamma \subseteq (\mathcal{O} \to \mathcal{A}) \to \mathbb{B}$ be a space of goal predicates on policy functions. Let $\gamma \in \Gamma$ be a goal predicate. We assume that the policy space $\Pi \subseteq \mathcal{O} \to \mathcal{A}$ can be sampled from, i.e., $\pi \sim \Pi$ returns a random element from Π. Random search for n samples is given via the function

$$\rho(n) = \begin{cases} \emptyset & \text{if } n = 0, \\ \pi & \text{if } n > 0 \text{ and } \gamma(\pi) \text{ where } \pi \sim \Pi, \\ \rho(n-1) & \text{otherwise.} \end{cases}$$

Definition 2 (optimization)**.** Let $\mathcal{X}$ be an arbitrary state space. Let $\mathcal{T}$ be an arbitrary set called target space and let $\leq$ be a total order on $\mathcal{T}$. A total function $\tau : \mathcal{X} \to \mathcal{T}$ is called target function. Optimization (minimization/maximization) is the procedure of searching for an $x \in \mathcal{X}$ so that $\tau(x)$ is optimal (minimal/maximal). Unless stated otherwise, we assume minimization. An optimization run of length $g+1$ is a sequence of states $\langle x_t \rangle_{0 \leq t \leq g}$ with $x_t \in \mathcal{X}$ for all t.

Let $e : \langle \mathcal{X} \rangle \times (\mathcal{X} \to \mathcal{T}) \to \mathcal{X}$ be a possibly randomized or non-deterministic function so that the optimization run $\langle x_t \rangle_{0 \leq t \leq g}$ is produced by calling e repeatedly, i.e., $x_{t+1} = e(\langle x_u \rangle_{0 \leq u \leq t}, \tau)$ for all t, $1 \leq t \leq g$, where x_0 is given externally (e.g., $x_0 =_{def} 42$) or chosen randomly (e.g., $x_0 \sim \mathcal{X}$). An optimization process is a tuple $(\mathcal{X}, \mathcal{T}, \tau, e, \langle x_t \rangle_{0 \leq t \leq g})$.

Definition 3 (optimization (policy))**.** Let $\mathcal{X} = \Pi$ be a policy space. Let $\mathcal{D} = (\Pi, \mathcal{T}, \tau, e, \langle x_t \rangle_{0 \leq t \leq g})$ be an optimization process according to Definition 2. $\mathcal{D}$ is called a policy optimization process.

Algorithm 3 (simulated annealing)**.** Let $\mathcal{D} = (\mathcal{X}, \mathcal{T}, \tau, e, \langle x_u \rangle_{0 \leq u \leq t})$ be an optimization process. Let *neighbors* : $\mathcal{X} \rightarrow \wp(\mathcal{X})$ be a function that returns a set of neighbors of a given state $x \in \mathcal{X}$. Let $\kappa : \mathbb{N} \rightarrow \mathbb{R}$ be a temperature schedule, i.e., a function that returns a temperature value for each time step. Let $A : \mathcal{T} \times \mathcal{T} \times \mathbb{R} \rightarrow \mathbb{P}$ with $\mathbb{P} = [0; 1] \subset \mathbb{R}$ be a function that returns an acceptance probability given two target values and a temperature. Commonly, we use

$$A(Q, Q', K) = e^{\frac{-(Q'-Q)}{K}}$$

for $\mathcal{T} \subseteq \mathbb{R}$. The process $\mathcal{D}$ continues via simulated annealing if e is of the form

$$e(\langle x_u \rangle_{0 \leq u \leq t}, \tau) = x_{t+1} = \begin{cases} x'_t & \text{if } \tau(x'_t) \leq \tau(x_t) \text{ or } r \leq A\big(\tau(x_t), \tau(x'_t), \kappa(t)\big), \\ x_t & \text{otherwise,} \end{cases}$$

where $x'_t \sim \textit{neighbors}(x_t)$ and $r \sim \mathbb{P}$ are drawn at random for each call to e.

Definition 4 (population-based optimization)**.** Let $\mathcal{X}$ be a state space. Let $\mathcal{T}$ be a target space with total order $\leq$. Let $\tau : \mathcal{X} \rightarrow \mathcal{T}$ be a target function. A tuple $\mathcal{E} = (\mathcal{X}, \mathcal{T}, \tau, E, \langle X_t \rangle_{0 \leq t \leq g})$ is a population-based optimization process iff $X_t \in \wp^*(\mathcal{X})$ for all t and $E : \langle \wp^*(\mathcal{X}) \rangle \times (\mathcal{X} \rightarrow \mathcal{T}) \rightarrow \wp^*(\mathcal{X})$ is a possibly randomized, non-deterministic, or further parametrized function so that the population-based optimization run is produced by calling E repeatedly, i.e., $X_{t+1} = E(\langle X_u \rangle_{0 \leq u \leq t}, \tau)$ where X_0 is given externally or chosen randomly.

Definition 5 (population-based optimization (alternate))**.** An optimization process $\mathcal{E} = (\mathcal{X}, \mathcal{T}, \tau, E, \langle X_t \rangle_{0 \leq t \leq g})$ is called population-based iff $\mathcal{X}$ has the form $\mathcal{X} = \wp^*(\mathcal{Y})$ for some other state space $\mathcal{Y}$.

Algorithm 4 (basic evolutionary algorithm)**.** Let $\mathcal{E} = (\mathcal{X}, \mathcal{T}, \tau, E, \langle X_u \rangle_{0 \leq u \leq t})$ be a population-based optimization process. The process $\mathcal{E}$ continues via an evolutionary algorithm if E has the form

$$E(\langle X_u \rangle_{0 \leq u \leq t}, \tau) = X_{t+1} = \textit{selection}\big(X_t \uplus \textit{variation}(X_t)\big)$$

where *selection* and *variation* are possibly randomized or non-deterministic functions so that for any $X \in \wp^*(\mathcal{X})$ it holds that $|\textit{selection}(X)| \leq |X|$ and $\big|\textit{selection}\big(X \uplus \textit{variation}(X)\big)\big| = |X|$.

Theorem 1 (no free lunch [1, 2])**.** As measured by sample efficiency, i.e., the achieved minimal value of τ per evaluations of $\tau(x)$ for some new $x \in \mathcal{X}$ for finite $\mathcal{X}$, all optimization algorithms perform the same when averaged over all possible target functions τ. So, for any search/optimization algorithm, any elevated performance over one class of problems is exactly paid for in performance over another class.

Definition 6 (multi-objective optimization)**.** Let $\mathcal{E} = (\mathcal{X}, \mathcal{T}, \tau, E, \langle X_u \rangle_{0 \leq u \leq t})$ be an optimization process. $\mathcal{E}$ is a multi-objective optimization process iff the target space $\mathcal{T}$ has the form $\mathcal{T} = \mathcal{T}_0 \times \cdots \times \mathcal{T}_{N-1}$ for some $N \in \mathbb{N}$ with $<_i$ being a total order on $\mathcal{T}_i$ for any $i \in [0; N-1] \subset \mathbb{N}$. Unless stated otherwise, we assume that no single total order on $\mathcal{T}$ is available. However, we can construct a partial order $\prec$ between target values $g, g' \in \mathcal{T}$ so that

$$(g_0, ..., g_{N-1}) \prec (g'_0, ..., g'_{N-1}) \iff \forall i \in [0; N-1] \subset \mathbb{N} \; : \; g_i < g'_i,$$

which is sufficient to adapt many standard optimization algorithms.

Definition 7 (Pareto front for optimization)**.** Let $\mathcal{E} = (\mathcal{X}, \mathcal{T}, \tau, E, \langle X_u \rangle_{0 \leq u \leq t})$ be a multi-objective optimization process with $\prec$ being a partial order on the multi-objective target space $\mathcal{T}$.

- A solution candidate x Pareto-dominates a solution candidate x', written $x \prec x'$ (assuming minimization), iff $\tau(x) \prec \tau(x')$.
- A solution candidate x is Pareto-optimal if there exists no other solution candidate $x' \in \mathcal{X}$ so that $x' \prec x$.
- The set of all Pareto-optimal solution candidates in $\mathcal{X}$ is called the Pareto front of $\mathcal{X}$ (w.r.t. $\prec$).

Algorithm 5 (gradient descent)**.** Let $\mathcal{E} = (\mathcal{X}, \mathcal{T}, \tau, e, \langle x_u \rangle_{0 \leq u \leq t})$ be an optimization process. Let $\mathcal{T}$ be continuous ($\mathcal{T} = \mathbb{R}$, e.g.) and let $\tau' : \mathcal{X} \to \mathcal{T}$ be the first derivative of τ. The process $\mathcal{E}$ continues via gradient descent (with learning rate $\alpha \in \mathbb{R}^+$) if e is of the form

$$e(\langle x_u \rangle_{0 \leq u \leq t}, \tau) = x_{t+1} = x_t - \alpha \cdot \tau'(x_t).$$

The learning rate α can also be given as a function, usually $\alpha : \mathbb{N} \to \mathbb{R}$, so that $e(\langle x_u \rangle_{0 \leq u \leq t}, \tau) = x_{t+1} = x_t - \alpha(t) \cdot \tau'(x_t)$.
If the computation of τ' is stochastic, usually because it is derived from a stochastic sampling of data points, this process is called stochastic gradient descent (SGD).

Algorithm 6 (gradient descent (policy))**.** Let π_θ be a policy π that depends on a vector of continuous parameters $\theta \in \Theta$, usually with $\Theta = \mathbb{R}^N$ for some N. Let $\tau : \Theta \to \mathcal{T}$ be a target function on the parameters θ of a policy π_θ. Let $\mathcal{T}$ be continuous ($\mathcal{T} = \mathbb{R}$, e.g.) and let $\tau' : \Theta \to \mathcal{T}$ be the first derivative of τ, i.e., $\tau'(\theta) = \frac{\partial \, \tau(\theta)}{\partial \, \theta}$. If $\mathcal{E} = (\Theta, \mathcal{T}, \tau, e, \langle x_u \rangle_{0 \leq u \leq t})$ is an optimization process that continues via gradient descent, $\mathcal{E}$ is a process of policy optimization via gradient descent.

References

[1] No free lunch theorems. `http://www.no-free-lunch.org`. Accessed 2022-05-10.

[2] David H Wolpert and William G Macready. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82, 1997.