

# Dokumentation der Simulated Annealing Implementierung

11. Juli 2025

## Dokumentation der `main.c`

Die Datei `main.c` enthält die vollständige Implementierung der Metaheuristik Simulated Annealing zur Optimierung der Reihenfolge von Fräsarbeitsgängen. Im Folgenden werden alle Funktionen und deren Zweck kurz beschrieben.

`name_to_idx` Wandelt einen Knotennamen (`char*`) in den zugehörigen Index im Knotenarray um. Wird für die Zuordnung von Kanten und Knoten verwendet.

`graphml_to_adjacency_matrix` Liest eine GraphML-Datei ein, zählt die Anzahl der Knoten und Kanten, speichert die Knotennamen und füllt die Adjazenzmatrix mit den Kantengewichten. Initialisiert außerdem die globale Knotengewichtsumme um sie nicht ständig neu zu berechnen.

`calculate_cost` Berechnet die Kosten eines Pfades, indem die Kantengewichte aufsummiert werden. Ungültige Übergänge werden mit einer hohen Strafe versehen. Die Knotengewichte werden zu den Gesamtkosten hinzugefügt.

`random_swap` Vertauscht zufällig zwei Elemente im Pfad. Dient der Erzeugung neuer Nachbarschaftslösungen im Simulated Annealing. Erzeugt potentiell Löcher im Pfad, welche durch `maximise_path` behoben werden.

`maximise_path` Versucht, einen Pfad möglichst gültig zu machen, indem ungültige Übergänge durch gezielte Vertauschungen minimiert werden.

`get_valid_transition_count` Gibt die Anzahl gültiger Übergänge (Kanten mit Gewicht ungleich Null) im Pfad zurück. Wird zur Bewertung der Pfadqualität genutzt.

`simulated_annealing` Implementiert den Simulated Annealing Algorithmus. Mutiert den Pfad iterativ durch zufällige Vertauschungen und akzeptiert neue Lösungen nach dem Metropolis-Kriterium. Gibt regelmäßig Zwischenstände und eine ETA aus.

**generate\_random\_path** Erzeugt eine zufällige Permutation der Knoten als Startpfad für die Optimierung.

**write\_best\_path\_txt** Schreibt den besten gefundenen Pfad als Semikolon-separierte Liste in eine Textdatei. Die Datei wird im Ordner `_bestpath_graph` abgelegt und enthält zusätzlich die Kosten des Pfades.

**main** Hauptfunktion. Liest die Eingabedatei, initialisiert die Datenstrukturen, gibt die Adjazenzmatrix aus, führt mehrere Simulated Annealing Durchläufe aus um tatsächlich gebrauch von den Temperaturen zu machen, wählt den besten Pfad dieser Läufe und gibt diesen sowohl auf der Konsole als auch in einer Textdatei aus. Am Ende werden alle Ressourcen freigegeben.