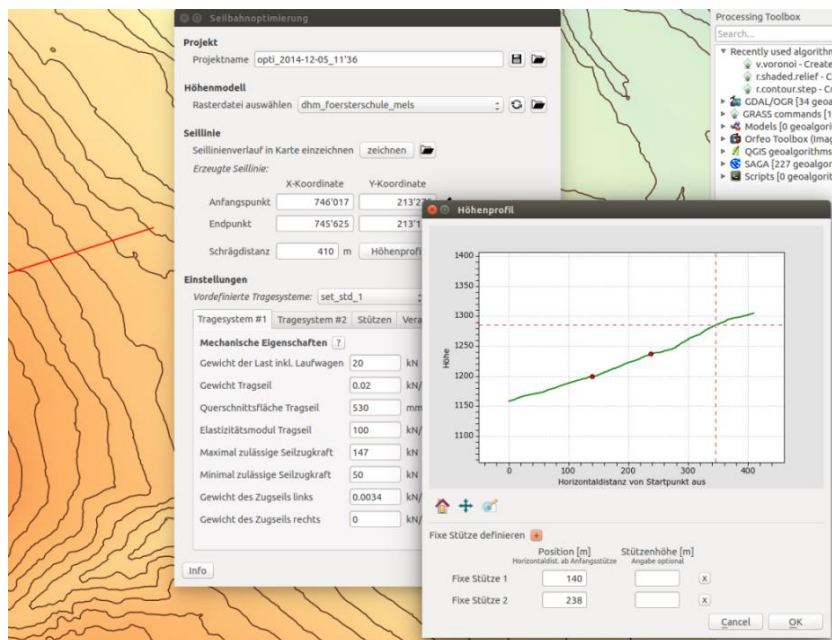


Dokumentation SEILAPLAN V2.0

Q-GIS Plugin zur Planung von Seilkran-Layouts



Inhaltsverzeichnis

1.	Zweck des Plugins.....	2
2.	Funktionsumfang und Bedienung.....	2
2.1.	Kompatibilität	2
2.2.	Installation.....	2
2.3.	Deinstallation	3
2.4.	Problembehebung	3
2.5.	Bedienung des Plugins	3
2.6.	Output.....	5
3.	Aufbau des Skripts	5
3.1.	Einbettung in Q-GIS: GUI und Hilfsfunktionen	7
3.2.	Kerncode: Optimierungsalgorithmus.....	8
3.3.	Output: Darstellung der Resultate und Erzeugen der Geodaten	9
4.	Code bearbeiten.....	9
4.1.	Allgemeines zur Arbeit mit dem Code.....	9
4.2.	Debugging mit PyCharm und QGIS.....	9
4.3.	Code auf GitHub verwalten und versionieren	10
4.4.	Weiterentwicklungen und To-Do-Liste	10

1. Zweck des Plugins

Das Plugin implementiert Leo Bonts Optimierungs-Algorithmus zur Planung eines forstlichen Seilkran-Layouts. Das Programm ist fähig, auf Grund eines digitalen Höhenmodells (DEM) zwischen definierten Anfangs- und Endkoordinaten das optimale Seillinienlayout zu berechnen (Position und Höhe der Stützen).

Das Programm ist für mitteleuropäische Verhältnisse konzipiert und geht von einem an beiden Enden fix verankerten Trageseil aus. Für die Berechnung der Eigenschaften der Lastwegkurve wird eine iterative Methode verwendet, welche von Zweifel (1960) beschrieben und speziell für an beiden Enden fix verankerten Trageseilen entwickelt wurde. Bei der Prüfung der Machbarkeit der Seillinie wird darauf geachtet, dass 1) die maximal zulässigen Spannungen im Trageseil nicht überschritten werden, 2) ein minimaler Abstand zwischen dem Trageseil und dem Untergrund gegeben ist und 3) bei einem Einsatz eines Gravitationssystems eine minimale Neigung im Trageseil gegeben ist. Es wird diejenige Lösung gesucht, welche in erster Priorität eine minimale Anzahl an Stützen aufweist und in zweiter Priorität die Stützenhöhe minimiert.

Die Einbettung im kostenlos erhältlichen Geografischen Informationssystem QGIS <http://qgis.com/> ermöglicht es, räumliche Daten zur Planung (Höhenmodell, Seillinie und Stützenpunkte) auf einer Karte darzustellen und die Geo-Funktionalität des GIS ins Tool einzubeziehen. So ist es beispielsweise möglich, die Seillinie in der Karte einzuzeichnen und sich das Höhenprofil anzeigen zu lassen.

2. Funktionsumfang und Bedienung

Für die Ausführung des Optimierungs-Algorithmus steht eine grafische Benutzeroberfläche (GUI = graphic user interface) zur Verfügung, mit deren Hilfe Start- und Endpunkt der Seillinie, die Eigenschaften des Tragesystems und weitere Einstellungsmöglichkeiten vom Benutzer spezifiziert werden können.

2.1. Kompatibilität

Das Plugin wurde unter Linux entwickelt und auf Windows getestet. Kompatibilität mit Mac-Systemen wurde nicht getestet und kann deshalb nicht gewährleistet werden.

Das Plugin wurde ursprünglich unter QGIS Version 2.6 und 2.8 entwickelt und im Nachhinein auf QGIS Version 3.4 portiert. Das Plugin sollte unter allen 3er Version von QGIS lauffähig sein. Da der Wechsel auf Version 3 grosse Änderungen in der Python API mit sich gebracht hat, ist die Unterstützung früherer Versionen nicht mehr möglich.

2.2. Installation

Informationen zur Installation von QGIS sind [auf http://qgis.com/](http://qgis.com/) zu finden.

Das Plugins wird in QGIS über das Erweiterungs-Menü hinzugefügt. In *Erweiterungen verwalten und installieren...* kann unter *Einstellungen* ein neues Repositorium hinzugefügt werden. Die Adresse des Repositoriums lautet

<https://raw.githubusercontent.com/piMoll/SEILAPLAN/master/plugin.xml>



Abb. 1
SEILAPLAN Icon

Im Feld *Namen* kann jeglicher Name eingesetzt werden. Nach Bestätigung des Dialogfensters kann im Reiter *Nicht installiert* das Plugin per Suchabfrage („SEILAPLAN“) gefunden und installiert werden. Bei der Installation kann es vorkommen, dass QGIS für eine kurze Zeit einfriert. Das Plugin Icon (Abb. 1) sollte nun im oberen Teil der QGIS Bedienoberfläche erscheinen. Falls nicht, kann per Rechtsklick auf die Oberfläche und Aktivierung des Eintrags *Werkzeugkästen > Erweiterungswerkzeuge* die aktivierten Plugins sichtbar gemacht werden.

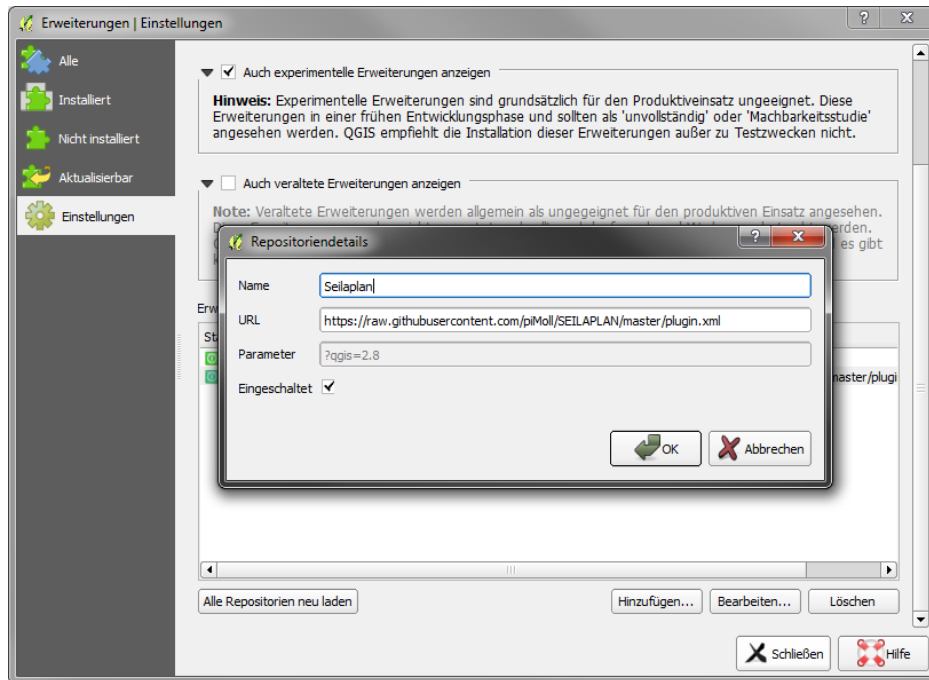


Abb. 2 Repository hinzufügen

2.3. Deinstallation


Das Plugin kann über das QGIS Menü *Erweiterungen verwalten und installieren...* deaktiviert oder deinstalliert werden. Im Reiter *Installiert* den Eintrag SEILAPLAN auswählen und auf deaktivieren, bzw. deinstallieren klicken.

2.4. Problembehebung

War die Installation oder Deinstallation nicht erfolgreich oder sind plötzlich zwei Plugins installiert, sollten die Plugin-Daten gelöscht und anschliessende eine Neuinstallation durchgeführt werden. Sofern keine speziellen QGIS-Profiles angelegt wurden, befinden sich Plugins unter Windows im Ordner `C:\Users\<benutzername>\AppData\Roaming\QGIS\QGIS3\profiles\default\python\plugins`, *benutzername* muss mit dem korrekten Windows User ersetzt werden. Der Plugin-Ordner SEILAPLAN kann gelöscht werden, QGIS sollte dabei jedoch nicht laufen. Anschliessend QGIS öffnen und SEILAPLAN erneut installieren. Unter Linux sind die QGIS Plugins in `/home/<USER>/.local/share/QGIS/QGIS3/profiles/default/python/plugins` zu finden (standardmässig versteckt).



2.5. Bedienung des Plugins

Per Klick auf das SEILAPLAN Icon wird das Hauptfenster des Plugins geöffnet. Das Plugin zeigt bereits einen aus Datum und Uhrzeit generierten Projektnamen und die im QGIS Projekt vorhandenen Rasterdaten an. Beide Werte können angepasst werden.

Höhenmodell: Sind im Projekt keine Rasterdaten vorhanden, können kein Anfangs- und Endpunkt der Seillinie definiert und das Plugin nicht ausgeführt werden. Wird ein Raster zu QGIS hinzugefügt, kann über die Schaltfläche  die Liste der Raster aktualisieren. Als Höhenmodell sind alle gängigen Rasterformate möglich, die von QGIS unterstützt werden. Geprüft wurde das Tool mit TXT-, TIF- und ASC-Rasterdateien. ArcGIS Raster innerhalb einer Geodatabase müssen erst in eines der genannten Formate exportiert werden damit sie in QGIS geladen und vom Plugin verarbeitet werden können.

Verlauf der Seillinie: Der Algorithmus basiert darauf, dass die Anfangs- und Endstütze in Form von Koordinaten vom Benutzer definiert werden. Der Startpunkt ist derjenige Punkt, auf welchem sich das Prozessorkippmastgerät befindet (Installationsplatz / Position des Masts) und welcher zuerst in der Karte gezeichnet wird. Koordinaten können entweder von Hand in die Felder geschrieben oder mit dem Zeichnen-Werkzeug direkt im GIS eingezeichnet werden.

Seillinienverlauf in Karte einzeichnen

Bei der Wahl von Anfangs- und Endpunkt muss beachtet werden, dass ein gewisser Höhenunterschied nötig ist und dass das Gelände nicht zu viele Gegensteigungen besitzt. Sind diese Voraussetzungen nicht erfüllt, kann keine Seillinie berechnet werden. Es wird deshalb empfohlen, zusätzliche GIS-Layer beizuziehen, die den Geländeverlauf besser visualisieren. Gut geeignet sind dafür Übersichtskarten oder Höhenlinien. Eine Übersichtskarte von Open Street Map kann über den Button  angezeigt werden. Höhenlinien können vom Plugin direkt aus dem Höhenmodell erzeugt und angezeigt werden, dazu den Button  klicken.

Eigenschaften der Seillinie/Parametersatz: Für die Eigenschaften der Seilbahn können vordefinierte Parametersätze gewählt werden und/oder von Hand eigene Werte definiert werden. Standardmässig müssen alle Parameterfelder gefüllt sein und die Werte müssen in einen vordefinierten Wertebereich liegen, ansonsten wird eine Fehlermeldung ausgegeben.

Variabelname	Standardwert	Bezeichnung	Einheit	dtype	min	max
Q	25	Gewicht der Last inkl. Laufwagen	kN	float	1	100
qT	0.0228	Gewicht Tragseil	kN/m	float	0.001	1
A	380	Querschnittsfläche Tragseil	mm ²	float	50	5000
E	100	Elastizitätsmodul Tragseil	kN/mm ²	float	10	1000
zul_SK	179	Maximal zulässige Seilzugkraft	kN	float	10	500
min_SK	50	Minimal zulässige Seilzugkraft	kN	float	10	zul_SK
qz1	0.0058	Gewicht des Zugseils links	kN/m	float	0	1
qz2	0	Gewicht des Zugseils rechts	kN/m	float	0	1
Bodenabst_min	7	Minimaler Abstand Tragseil - Boden	m	float	0	HM_min
Bodenabst_A	40	einzuhalten ab (vom Startpunkt)	m	int	0	200
Bodenabst_E	40	einzuhalten bis (vor dem Endpunkt)	m	int	0	200
GravSK	ja	Gravitationsseilkan?		string	nein	ja
Befahr_A	0	Befahrbarkeit ab (vom Startpunkt)	m	int	0	1000
Befahr_E	0	Befahrbarkeit bis (vor dem Endpunkt)	m	int	0	1000
HM_Anfang	11	Höhe der Anfangsstütze	m	int	0	30
d_Anker_A	10	Länge der Verankerung	m	float	0	100
HM_Ende_min	0	Minimale Höhe der Endstütze	m	int	0	HM_max
HM_Ende_max	10	Maximale Höhe der Endstütze	m	int	0	HM_max
d_Anker_E	10	Länge der Verankerung	m	float	0	100
Min_Dist_Mast	10	Minimaler Abstand zwischen Stützen	m	int	0	200
L_Delta	10	Horiz. Auflösung mögl. Stützenstandorte	m	int	1	20
N_Zw_Mast_max	10	Maximale Anzahl Zwischenstützen		int	1	30
HM_min	8	Minimale Stützenhöhe	m	int	Bodenabst_min	HM_max
HM_max	18	Maximal Stützenhöhe	m	int	HM_min	30
HM_Delta	2	Abstufungsintervall	m	int	1	5
HM_nat	14	Künstliche Stützen ab Stützenhöhe	m	int	0	100
A_SK	12	Anfangsseilkraft	kN	float	1	500
Min_Gradient	0	Min. nötiger Gradient		float	0	inf
Federkonstante	inf	Federkonstante der Verankerung	kN/m	float	0	inf
L_min	30	Minimale Länge eines Seilfeldes	m	float	0	100

2.6. Output

Das Plugin kann unterschiedliche Arten von Outputdaten erzeugen, die per Klick auf das Feld *Output Optionen* ausgewählt werden können. Standardmässig wird ein Plot mit dem Seillinienverlauf und den berechneten Stützen generiert und ein ausführlicher technischer Bericht, der die Kennwerte der Seillinie auflistet. Sind weitere Analysen mit Geodaten nötig, kann das Plugin zusätzlich Shapedaten erzeugen oder CSV-Tabellen mit Koordinaten ausgeben. Unter *Speicherpfad* kann auch der Ausgabeordner definiert werden. Jede Durchführung des Algorithmus erzeugt im genannten Ordner einen neuen Ordner, der gleich wie das Projekt heisst.

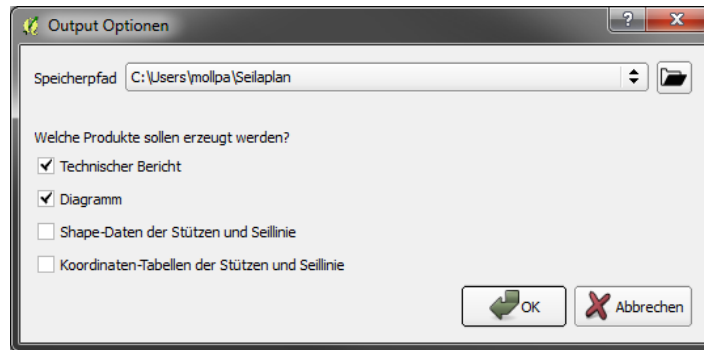


Abb. 3 Output Optionen

3. Aufbau des Skripts

Die offizielle QGIS Seite beschreibt die Plugin Grundbausteine auf dieser Seite: http://docs.qgis.org/testing/en/docs/pyqgis_developer_cookbook/plugins.html

Nachstehend sind alle wichtigen Dateien im Skript-Verzeichnis aufgelistet (nicht abschliessend). In den einzelnen Kapiteln folgt eine genauere Erklärung welche Skripts für was zuständig sind und auf welche Weise sie einander aufrufen.

Ordner	Skriptdatei	Beschreibung
bo		Hilfsklassen, welche für die Funktionalität der GUI innerhalb QGIS nötig sind. Sämtliche Scripts in diesem Ordner stammen aus einem anderen Plugin ..., welches ähnliche Funktionalitäten aufweist wie das SEILAPLAN Plugin.
config		Ordner mit Parameter-Dateien.
	params.txt	Enthält Parametersätze für unterschiedliche Tragesysteme-Konfigurationen. Enthalten sind nebst den Werten auch genaue Erklärungen, Wertebereiche, Datentypen und Einheiten, die zur Prüfung der Werte verwendet werden.
	commonPaths.txt	Enthält die vom Benutzer gewählten Output Pfade und die Einstellungen für die Output Optionen.
gui		Enthält Skripte, die die Darstellung der GUI und ihr Verhalten definieren.
	guiHelperFunctions.py	weitere Funktionen der HauptGUI, die das File seilaplanPluginDialog.py ergänzen.
	ui_seilaplanDialog.py	Alle grafischen Elemente der HauptGUI.
	profileDialog.py	Funktionen und grafische Elemente des Höhenprofil-Dialogs.
	profilePlot.py	Graph, der das Höhenprofil anzeigt.
	progressDialog.py	Fortschrittsfenster, welches während der Berechnungen angezeigt wird
	seilaplanDialog.ui	Gespeichertes QTDesigner -Projekt mit HauptGUI.
help		
i18n		

icons		Icons und Hilfsgrafiken, die von der HauptGUI angezeigt werden.
img		Beschreibende Darstellungen für Parameter-Definition.
tool		Sämtliche Scriptdateien des Kernalgorithmus.
	mainSeilaplan.py	Hauptsript.
	geoExtract.py	Bestimmt das Bodenprofil durch Interpolation auf dem Höhenraster, berechnet die möglichen Stützenpositionen mit Berücksichtigung von Bodenabstand und Befahrbarkeit, bestimmt die Ankerparameter.
	peakdetect.py	Kleines Tool das Erhöhungen und Spitzen im Höhenprofil bestimmt.
	mainOpti.py	Hauptsript für die Optimierung.
	optiSTA.py	Berechnet den Bereich der Anfangsseilzugkraft STA, welcher für den betrachteten Abschnitt möglich ist.
	cableline.py	Berechnung der Seillinie basierend auf der Methode von Zweifel im Rahmen des Optimierungsdurchlaufes.
	cablelineFinal.py	Berechnet den Seilverlauf, die exakte Seilmechanik und eine Vielzahl von Kennwerten des Tragesystems nach Fertigstellung des Optimierungsprozesses.
	outputReport.py	Erstellt Ausgabeplot mit Seillinie und Höhenprofil und erzeugt den Bericht.
	outputGeo.py	Erstellt Shape-Daten und CSV-Koordinatentabellen.
seilaplanPlugin.py		Haupt-Plugin-Script. Initialisiert GUI und startet Kernalgorithmus.
seilaplanPluginDialog.py		Funktionen der HauptGUI, sehr ausführliches Skript.
processingThread.py		Übernimmt Inputdaten des Benutzers und startet einen zweiten Thread um den Optimierungsalgorithmus durchzuführen.
resources_rc.py		Übersetzung von resources.qrc in ein Python-Script.
resources.qrc		
metadata.txt		Beschreibt das Plugin kurz und enthält alle Kenndaten, welche für ein QGIS-Plugin angegeben werden müssen (Version, Autor, etc.)

Der grobe Ablauf der Plugin-Ausführung ist in Abb. 4 dargestellt.

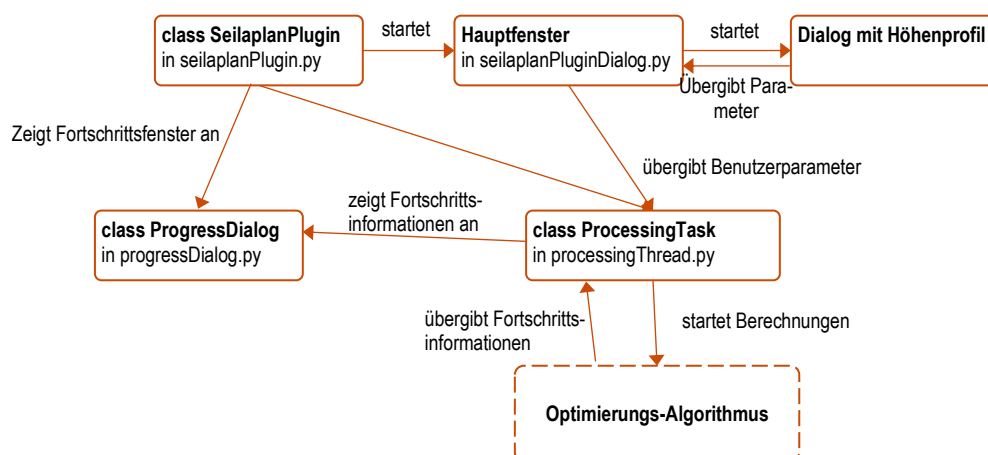


Abb. 4 Ablauf-Struktur des gesamten Plugins

3.1. Einbettung in QGIS: GUI und Hilfsfunktionen

Eine grosse Zahl von Python-Skripts ist zur Darstellung der GUI und der Funktionalität innerhalb QGIS nötig. Um das Plugin auszuführen, wird ein Dialog-Fenster initialisiert, das bei Klick auf OK den Optimierungs-Algorithmus ausführt. Rein organisatorisch basiert also der gesamte Algorithmus auf einer GUI-Instanz innerhalb QGIS, die die einzelnen Bestandteile des Skripts nacheinander aufruft.

Für Python gibt es unterschiedliche GUI-Bibliotheken wie WxPython, Tkinter oder PyQt. Da die Oberfläche von QGIS in Qt geschrieben ist (<http://qt-project.org>), muss auch für das Plugin-Design Qt und damit die Python Bindings PyQt verwendet werden (Bindings = Bibliotheken, die Funktionen anderer Software für Python nutzbar machen, beispielsweise arcpy für ArcGIS).

GUI-Design: Das Programmieren von GUIs basiert komplett auf Objektorientierung und der Einstieg ist eher anspruchsvoll. Etwas vereinfacht wird dies bei Qt dadurch, dass mit QtDesigner ein grafischer Editor zur Verfügung steht, mit dem das Aussehen der GUI erstellt werden kann. Die Positionierung von Labels, Feldern und Buttons ist damit sehr einfach, die Funktionen einzelner Buttons muss aber nachträglich noch programmiert werden.

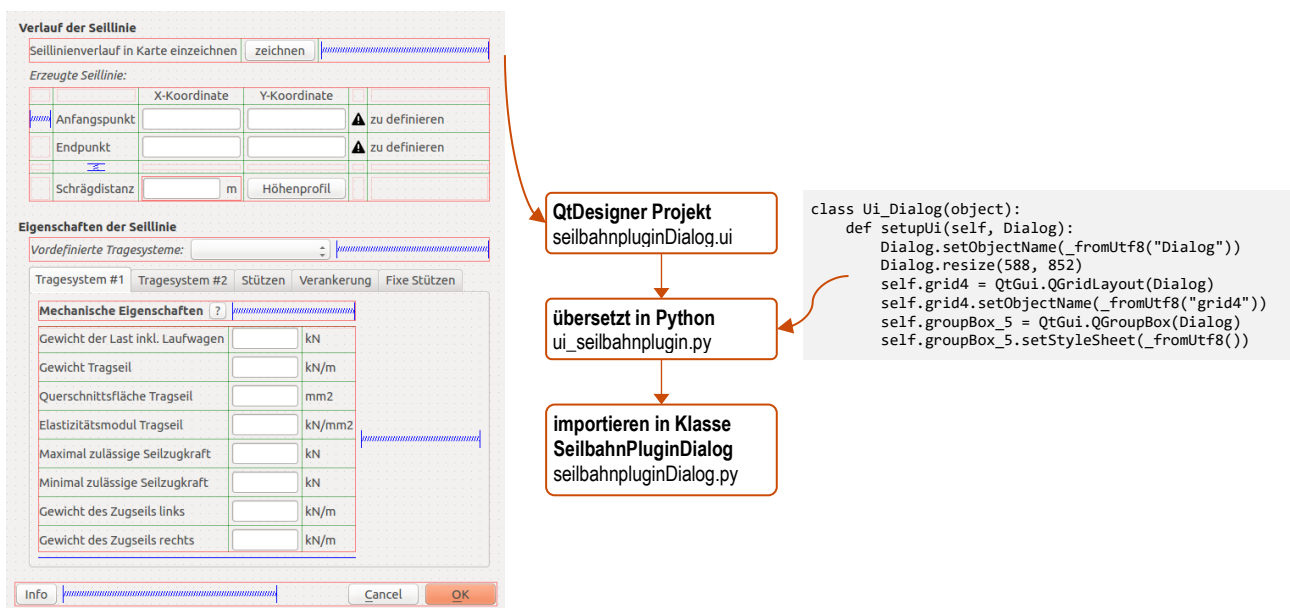


Abb. 5 Aussehen der GUI, erstellt in QtDesigner

Die Übersetzung des QtDesigner-Projektes in Python Code muss bei jeder Änderung der GUI über die Konsole ausgeführt werden:

1. Konsole öffnen und mit Befehl `cd` zu Ordner wechseln, der die ui-Datei enthält:
`cd /Path/to/Folder/SEILAPLAN/gui/`
2. Übersetzung mit Befehl `pyuic4` ausführen:
`pyuic5 seilbahnpluginDialog.ui -o ui_seilbahnplugin.py`

GUI-Funktionalität: Ist das Aussehen der Benutzeroberfläche aufgebaut, muss die Funktionalität der Buttons und Felder programmiert werden. Nachstehend sind die wichtigsten Funktionskategorien der GUI und die verantwortlichen Methoden in der `seilaplanPluginDialog.py` aufgelistet:

- GUI initialisieren und vorbereiten
- In QGIS vorhanden Raster im Drop-Down-Menü anzeigen
- Linie in Karte zeichnen
- Koordinaten von Anfangs- und Endpunkt überprüfen und layouten
- Profile-Fenster anzeigen
- Auslesen der Benutzerwerte und Wertebereichsprüfung

- Abspeichern der Parameter und Übergabe an separaten Thread
- Projekt abspeichern und laden

Multithreading und starten des Optimierungs-Algorithmus: Würde man im Hauptskript `seilaplanPlugin.py` in der Methode `run` den Algorithmus ausführen, würde bei der länger dauernden Berechnung QGIS in der Zwischenzeit einfrieren. Damit QGIS aktiv und ansprechbar bleibt, wird der Optimierungs-Algorithmus separat von QGIS ausgeführt, dieses Vorgehen wird als Multithreading bezeichnet. Multithreading oder auch Mehrsträngigkeit bedeutet in der Informatik das gleichzeitige

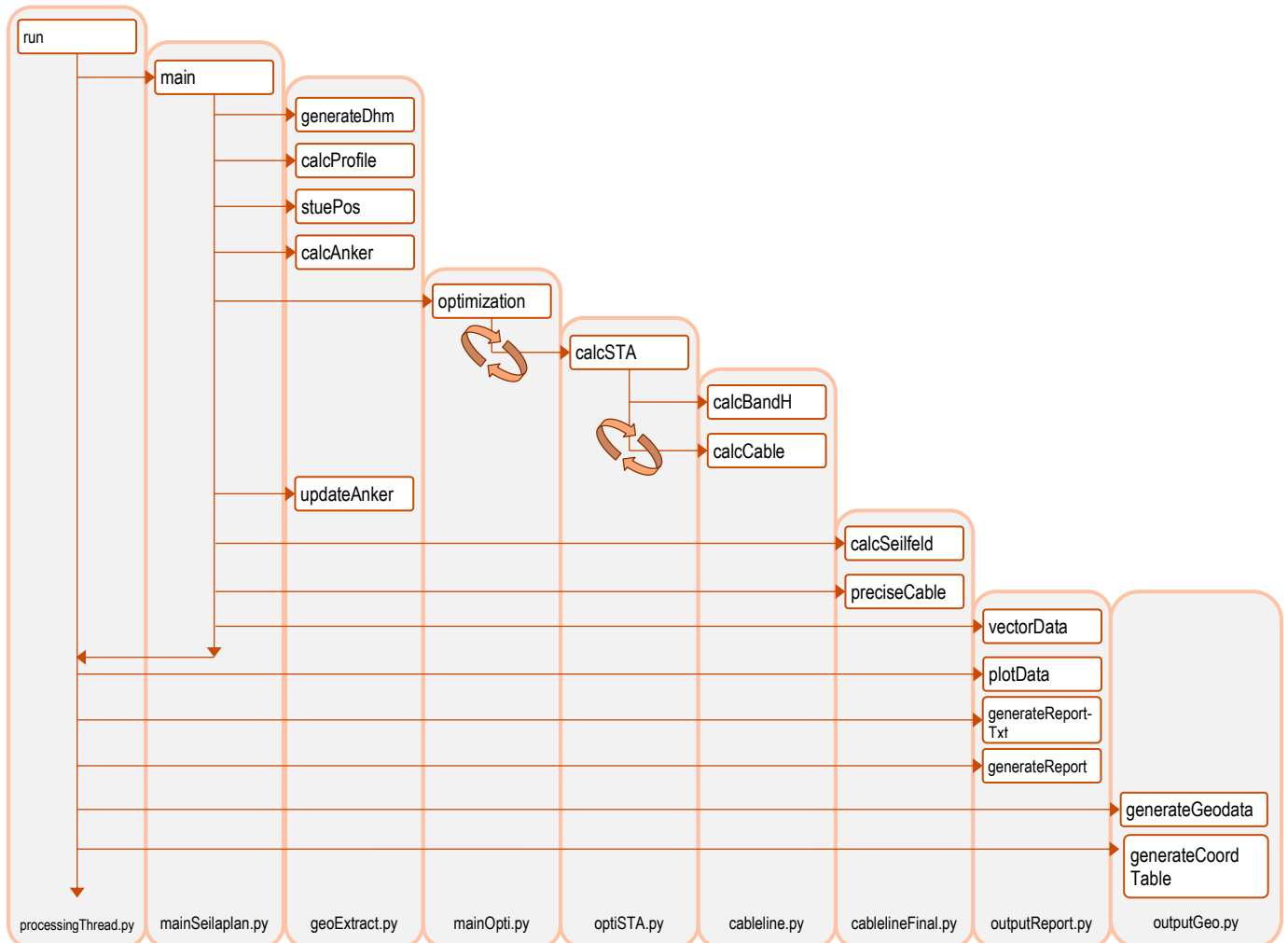


Abb. 6 Struktur des Kernalgorithmus. In den Kästchen sind einzelne Methoden angegeben, die grauen Boxen bezeichnen die jeweilige Python Skriptdatei.

Abarbeiten von mehreren Arbeitsprozessen innerhalb eines Programms. In der Skriptdatei `processingThread.py` wird dieser Thread eingerichtet und die Optimierung ausgeführt. Hier werden zudem die Rückgabeparameter des Algorithmus entgegengenommen und je nach Resultat (erfolgreich oder nicht) die weiteren Schritte ausgeführt.

3.2. Kerncode: Optimierungsalgorithmus

Die Abbildung unten zeigt die Abfolge des Algorithmus nach Bestätigung des Plugin-Dialogs. Die Methode `run` befindet sich im Skript `processingThread.py`.

Die Methodennamen entsprechen nicht denjenigen des ursprünglichen Matlab-Algorithmus, auch bei vielen Variablennamen wurde auf Grund der *Python Naming Convention* oder einfach zur besseren Lesbarkeit des Codes andere Bezeichnungen gewählt. Im Groben entsprechen die Matlab-Funktionen folgenden Python-Skripts:

Matlab Methode	Python Script
OptEinzSeillinie_ReVi....	mainSeilaplan.py, geoExtract.py
func_mastopt_aufgr_zul_SK_edit	mainOpti.py
EvalPotSTAZweifel	optiSTA.py
Eval_Cableline_Mastopt_Zweifel_zul_SK_only_one_feld	cabeline.py
CalculateSeilfeld	cablelinePrecise.py

3.3. Output: Darstellung der Resultate und Erzeugen der Geodaten

Für das Erzeugen und Darstellen der Resultate sind die Methoden in den Scripts *outputReport.py* und *outputGeo.py* zuständig.

Plot erstellen: In der Methode *plotData* werden die Seildaten, das Höhenprofil und weitere grafische Elemente in ein Diagramm gezeichnet. Vorgängig werden die Daten in der Methode *vectorData* so umgewandelt, dass sie korrekt dargestellt werden können.

Report erstellen: Der Text für den Bericht wird in der Methode *generateReportTxt* erstellt. Für jeden Abschnitt im PDF werden hier Bezeichnungen und Werte in Listen gespeichert. Das resultierende PDF wird in der Methode *generateReport* erstellt. Es besteht aus vielen einzelnen Tabellen, die mit Text gefüllt und gestylt werden.

Geodaten erstellen: Die Koordinatenlisten der Seillinie werden an die Methode *generateGeodata* übergeben, die je ein Shapefile für das Leerseil, das Lastseil und die Stützenpunkte erzeugt. In der Methode *generateCoordTable* werden die csv-Tabellen erzeugt. Die Seildaten enthalten die Horizontalabstand ab Anfangspunkt, X- und Y-Koordinaten, die Höhe des Last- und Leerseils, sowie das Höhenprofil. Die Stützeninformationen enthalten eine Bezeichnung, X- und Y-Koordinaten, die Z-Koordinate am Boden und an der Spitze der Stütze sowie die Höhe.

4. Code bearbeiten

4.1. Allgemeines zur Arbeit mit dem Code

Wenn Code in einem Plugin verändert wird, werden diese Änderungen erst mit einem Neustart von QGIS umgesetzt. Um das dauernde Neustarten zu umgehen, gibt es ein separates QGIS Plugin, welches das Neuladen des Codes übernimmt. Es heisst *Plugin Reloader* und kann über das Plugin Menü installiert werden.

Wenn bereits beim Start von QGIS oder beim Neuladen mit dem Plugin Reloader eine Fehlermeldung gezeigt wird, ist der Ursprung des Fehlers meist einer der folgenden drei Möglichkeiten: eine der vielen `__init__` Methoden ist fehlerhaft, ein Import ist falsch oder im Code gibt es ein Syntaxfehler.

4.2. Debugging mit PyCharm und QGIS

Debugging ist mittels PyCharm Remote Debugging möglich (nur in PyCharm Professional enthalten). So weit bekannt, gibt es sonst nicht viele andere Python-IDEs, die dies können. Die Einrichtungsanweisungen sind in PyCharm selbst unter „Remote Debugging“ gut erklärt. Nebst den Einstellungen in PyCharm sind folgende Einträge im Code nötig:

Am Anfang von *seilaplanPlugin.py* (bereits im Code vorhanden, jedoch auskommentiert)

```
try:
    import pydevd
    pydevd.settrace('localhost', port=53100, stdoutToServer=True, stderrToServer=True)
except ConnectionRefusedError:
```

```
pass
except ImportError:
    pass
```

Nach diesem Eintrag hält die Ausführung des Plugins an und man ist im Debug Modus. Anschliessend an diese Zeile können in PyCharm so viele Breakpoints wie gewünscht gesetzt werden und mit den internen Debug-Werkzeugen angesteuert werden. Ein Weiterer Eintrag ist für den separaten Berechnungs-Thread notwendig. In der Methode `run in processingThread.py` ist ebenfalls ein solcher Eintrag auskommentiert vorhanden.

4.3. Code auf GitHub verwalten und versionieren

Der Plugin Code wird auf dem Repository <https://github.com/piMoll/SEILAPLAN> gehostet und weiterentwickelt. Zur automatischen Verteilung neuer Versionen des Plugins sind folgende Schritte notwendig:

1. Versionsnummer im XML-Datei `plugin.xml` und `metadata.txt` erhöhen.
2. Commit und Push der aktualisierten und getesteten Code-Änderungen inkl. XML-Datei und `metadata.txt` (hochladen auf Github).
3. Per Download-Funktion auf GitHub das Plugin als ZIP-Datei herunterladen, entpacken, Ordnername von `SEILAPLAN-master` auf `SEILAPLAN` abändern und zu einem neuen Archiv `SEILAPLAN.zip` hinzufügen.
4. `SEILAPLAN.zip` auf Online Hosting Service hochladen.
5. Falls die aktualisierte Zip-Datei auf dem Hosting Service nicht mehr denselben Link aufweist wie in `plugin.xml` aufgeführt, muss der Link im XML angepasst und erneut auf Github hochgeladen werden.

Die manuelle Anpassung des Ordner- und Zip-Namens ist nötig, weil die in GitHub integrierte zip-Erzeugung und Download-Funktion nicht angepasst werden kann. Der automatisch generierte Zip-Name `SEILAPLAN-master` enthält das für QGIS Plugins ungültige `` Zeichen, weshalb eine manuelle Anpassung nötig ist.

Falls die automatische Überprüfung auf Updates in QGIS aktiviert wird, wird bei einem Neustart automatisch erkannt, dass eine neue Version des Plugins zum Download zur Verfügung steht. Per Klick auf die Benachrichtigung am unteren Rand wird das Plugin aktualisiert.

4.4. Weiterentwicklungen und To-Do-Liste

- Performanter Implementation des Yen K-shortest Path entwickeln
- Sauberer Benutzereingabe mit QValidator erstellen: <http://snorf.net/blog/2014/08/09/validating-user-input-in-pyqt4-using-qvalidator/>