

**Institut für Aufbau- und Verbindungstechnik
der Elektronik**

Fakultät Elektrotechnik und Informationstechnik

Technische Universität Dresden

**Erstellung eines Programminterfaces (Tcl/Matlab)
zur Analyse von Ergebnissen bei der
heuristischen Optimierung diskreter Fertigungsabläufe**

Dipl.-Math. Andreas Klemmt

15 September 2006

Zuständiger Projektleiter: PD Dr.-Ing. Gerald Weigert

Inhaltsverzeichnis

1	Einleitung	2
2	Programminterface (Tcl/Matlab)	2
3	Programmoberfläche Visualisierung_Matlab	7
3.1	Modelldaten	11
3.2	3D-Visualisierung des Optimierungsprozesses	11
3.3	Ansichtsoptionen	11
3.4	Korrelationskoeffizient	12
3.5	Visualisierungsparameter	13
3.6	Anzeigeoptionen	13
3.7	Bildexport	17
3.8	Statusfenster	19
3.9	Modell-Export/Import	20

1 Einleitung

Die Modellierung und Optimierung diskreter Fertigungsprozesse gewinnt zunehmend an Bedeutung. Es werden leistungsfähige Simulations- und Optimierungssysteme eingesetzt um die meist komplexen Zusammenhänge des Produktionsprozesses abzubilden.

Ziel des Einsatzes derartiger Systeme sind u.a. die Verringerung der mittleren Durchlauf- oder Maschinenstillstandszeit. Diese Zielgrößen hängen von Stellgrößen ab, welche in einem diskreten Steuerungsraum variiert werden können. Dieser Raum enthält insbesondere die Permutationen bestimmter Objekte. Für die Lösung von Reihenfolgeproblemen haben sich in der Praxis vor allem heuristische Optimierungsverfahren bewährt.

Ziel dieser Arbeit ist die Erstellung eines Programminterfaces, welches es ermöglicht, die Ergebnisse derartiger Optimierungsmethoden grafisch zu analysieren. Hierzu galt es für das Simulations- und Optimierungssystem *simcron MODELLER* eine Schnittstelle zu entwickeln, welche den Export der Optimierungsergebnisse in eine zu erstellende Matlab-Umgebung ermöglicht. Eine Beschreibung der Schnittstelle erfolgt anschließend in Sektion 2, jene der Matlab-Oberfläche in Sektion 3.

2 Programminterface (Tcl/Matlab)

Mit Hilfe der Skriptsprache Tcl ist ein Datenaustausch mit dem Simulations- und Optimierungssystem *simcron MODELLER* möglich. Die nachfolgende Tabelle 1 gibt einen Überblick über die, die Optimierung betreffenden Tcl-Komandos. Um diese Kommandos auszuführen, muss jedoch sichergestellt sein, dass das Optimierungspaket *Optx* geladen ist. Man benutze hierzu den Befehl:

```
package require Optx
```

Mit Hilfe der in Tabelle 1 erläuterten Kommandos wird im File *menu.tcl*, welches sich im Verzeichnis *'../Modeller3.2/lib/opt'* befindet, ein Matlab-File *Daten.m* erzeugt. Dieses dient als Datenquelle für die Matlab-Oberfläche, welche über das Programm *Matlab_Visualisierung* im Verzeichnis *'../Modeller3.2/lib/opt/matlab'* ausgeführt wird. Aufgrund der Tatsache, dass i.A. keine Schreibrechte für das Verzeichnis *../Modeller3.2/..* existieren, muss die Datei *Daten.m* in einem öffentlichen Ordner erstellt werden. Im vorliegenden Fall bot sich hierfür der temporäre Ordner *'../Lokale Einstellungen/Temp'* an.

Kommando	Erklärung
<code>OptAlgorithm</code>	Liefert String des verwendeten Optimierungsalgorithmuses.
<code>model origin</code>	Liefert String mit Pfad und Namen des Modells.
<code>OptAlgorithmConfigureList</code>	Liefert String mit den verwendeten Einstellungen des Optimierungsalgorithmuses.
<code>OptAdjustObjectTable</code>	Liefert eine geordnete Liste der Stellgrößen (Permutations- bzw. Adjustobjekte).
<code>OptGetConfigurationDiag</code>	Liefert eine geordnete Liste mit den Werten der Stellgrößen in jedem Iterationsschritt. Permutationen werden dabei als Liste ausgegeben.
<code>cost used</code>	Liefert eine geordnete Liste der Kostenobjekte (Zielgrößen).
<code>costX name</code>	Liefert einen String mit dem Namen der Zielgröße.
<code>costX optimize</code>	Liefert den Wert 1, wenn bzgl. dieser Zielgröße optimiert wird (sonst 0).
<code>OptGetOptimizationGraph</code>	Liefert eine geordnete Liste mit den Werten der Zielgrößen in jedem Iterationsschritt.
<code>OptGetOptimizationGraphAccepted</code>	Liefert eine Liste mit den vom Optimierer akzeptierten Iterationsschritten.

Tabelle 1: Tcl-Kommandos bzgl. Optimierung

Dieser existiert auf jedem Rechner und kann in Tcl mit Hilfe von

```
$::env(Temp)
```

bzw. in Matlab mit Hilfe von

```
tempdir
```

ausgelesen werden. Abbildung 1 erläutert den bisher gewonnenen Sachverhalt.

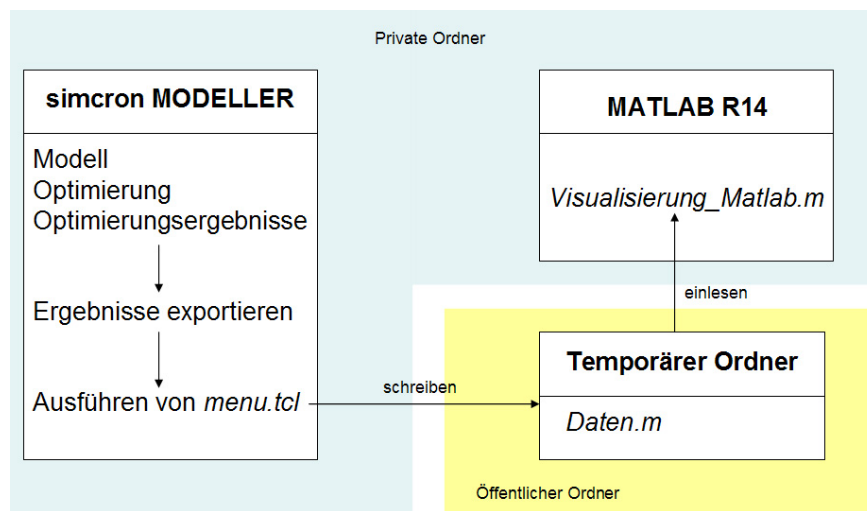


Abbildung 1: Veranschaulichung

Um dem Nutzer das manuelle Ausführen der Matlab-Oberfläche und das damit verbundene Einlesen des Files *daten.m* zu ersparen, wurde in *menu.tcl* die folgend dargestellte Execution-Anweisung implementiert.

```
exec $MatlabRoot /r "cd('$FileRoot'); run '$FileRoot2'" &
```

Diese startet die Ausführung des Programms *Visualisierung_Matlab.m*. Dabei enthält `$MatlabRoot` einen String mit dem Quellpfad von Matlab, `$FileRoot` einen String des Ordners `'../Modeller3.2/lib/opt/matlab'` und `$FileRoot2` einen String des auszuführenden Programms (im vorliegenden Fall) `'../Modeller3.2/lib/opt/matlab/Visualisierung_Matlab'`. Der Befehl `/r` macht es möglich eine auszuführende Anweisung an die Matlab-Konsole zu

übergeben, welche in `".."` eingeschlossen sein muss. Das Kommando `cd(..)` wechselt das aktuelle Matlab-Arbeitsverzeichnis und `run` führt das ihm nachstehende Matlab-Programm aus. Das `&` sorgt dafür, dass die Programme Matlab und `simcron MODELLER` parallel ausgeführt werden können.

Die Strings `$FileRoot` und `$FileRoot2` werden während der Ausführung von `menu.tcl` erzeugt. Dies ist möglich, da der Quellpfad des Programms `simcron MODELLER` zur Laufzeit bekannt ist. Anders hingegen ist dies beim Quellpfad `$MatlabRoot` von Matlab. Dieser muss dem Programm vom Nutzer übergeben werden. Um dies benutzerfreundlich zu gestalten, wurde das in Abbildung 3 dargestellte Popup implementiert, welches sich beim ersten Export von Optimierungsergebnissen (vgl. Abbildung 2) vom `simcron MODELLER` nach Matlab öffnet.

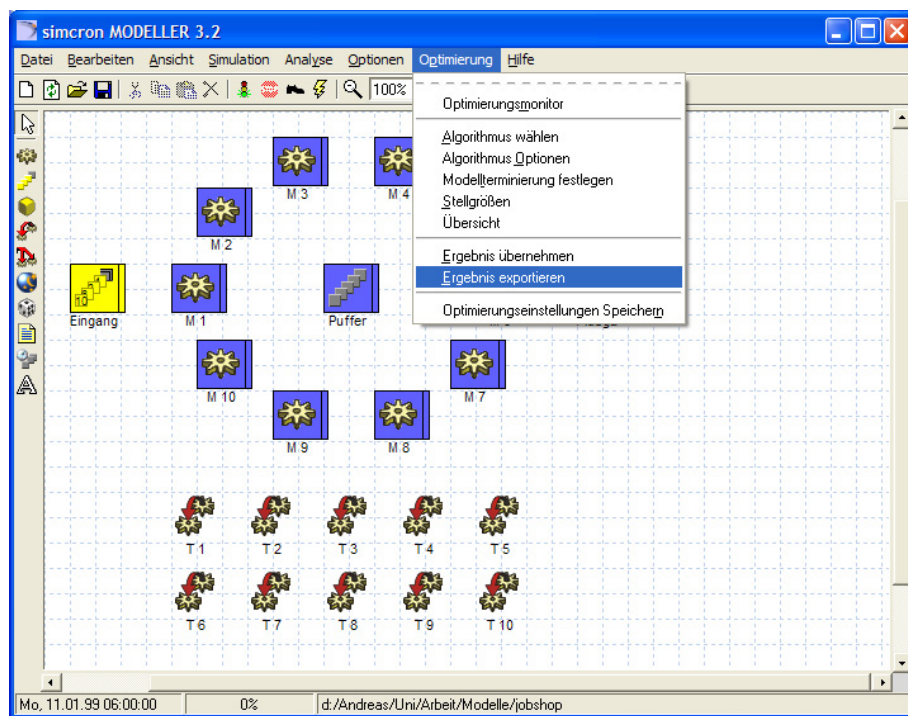


Abbildung 2: Exportieren der Modellierungsergebnisse

In diesem Popup ist der Matlab-Quellpfad auszuwählen (vgl. Abbildung 3). Die Unterverzeichnisse, in welchen sich die `matlab.exe` befindet, werden automatisch ergänzt. Damit dieses Popup nicht bei jedem Datenexport erscheint, wird der Matlab-Quellpfad als Umgebungsvariable `MATLAB_PFAD` im Sys-

tem abgelegt. Eine Abfrage im File *menu.tcl* prüft, ob diese Variable existiert und speichert deren Inhalt (bei Existenz) in `MatlabRoot` ab.



Abbildung 3: Popup

Sollte sich der Quellpfad von Matlab ändern (Neuinstallation von Matlab, Start über Server, lokale Installation), ist es notwendig, die im System abgelegte Umgebungsvariable `MATLAB_PFAD` zu löschen. Hierzu geht man wie folgt vor:

Start → Systemsteuerung → System → Erweitert → Umgebungsvariablen

Es zeigt sich das in Abbildung 4 dargestellte Fenster.

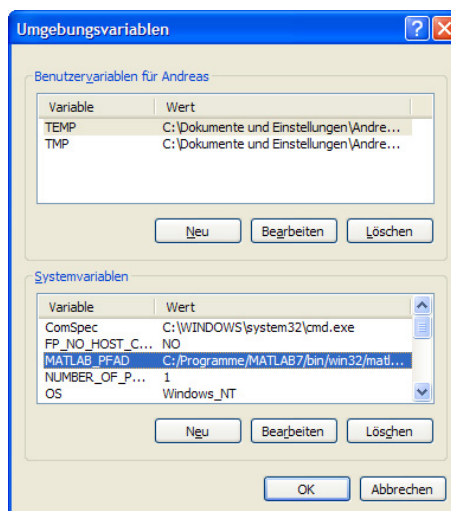


Abbildung 4: Löschen einer Umgebungsvariable

Nun ist das Markieren und Löschen der Umgebungsvariable möglich. Für weitere programmtechnische Details sei an dieser Stelle auf den Quellcode verwiesen.

3 Programmoberfläche Visualisierung_Matlab

Nachdem die Matlab/Tcl-Schnittstelle kurz vorgestellt wurde, widmet sich dieser Abschnitt der Erläuterung der Matlab Oberfläche. Diese wird, wie zuvor beschrieben, automatisch ausgeführt, sobald der Eintrag "Ergebnis exportieren" im Menü "Optimierung" des simcron MODELLER gewählt wurde (vgl. Abbildung 2). Es zeigt sich das in Abbildung 5 dargestellte GUI.

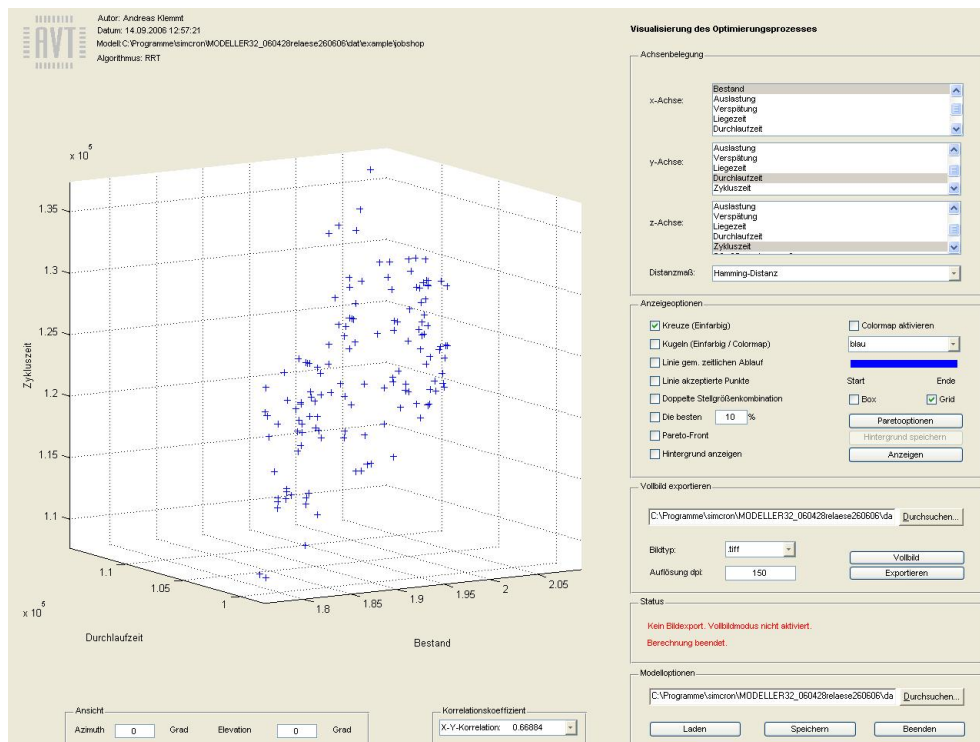


Abbildung 5: GUI Visualisierung_Matlab

Diese Oberfläche besteht insgesamt aus neun Dateien, welche sich im (festen) Ordner `'../Modeller3.2/lib/opt/matlab'` befinden. Tabelle 2 erläutert kurz diese Dateien sowie deren Funktion.

Datei	Erklärung
Visualisierung_Matlab.fig	Matlab-Figure der Oberfläche. Hier ist z.B. die Anordnung der Buttons, Listen und Checkboxes veränderbar. Hierzu ist der Befehl <code>guide</code> in der Matlab-Konsole einzugeben und die entsprechende Figur zu laden.
Visualisierung_Matlab.m	Implementierungsdatei der Matlab-Figure. Der gesamte Quellcode ist im Matlab-Editor einsehbar/editierbar.
AlterWindow.m	Hilfsfunktion welche es ermöglicht das Programm direkt im Vollbildmodus auszuführen.
AlterWindow.c, AlterWindow.mex32	Quelldateien für AlterWindow.m.
indexPareto.mat	Matlab-file zur Zwischenspeicherung bestimmter Variablen.
Tu_AVT.jpg	AVT-Logo.
Paretoauswahl.fig	Matlab-Figure der Paratoauswahloberfläche (vgl. 3.5).
Paretoauswahl.m	Zu Paretoauswahl.fig gehörende Implementierungsdatei.

Tabelle 2: Datein der Matlab-Oberfläche

Außerdem benötigt das Programm das bereits angesprochene Quelldatenfile *daten.m* aus dem öffentlichen, lokalen Ordner `'./Lokale Einstellungen/Temp'`. Dieses File initialisiert den Matlab-Workspace, während der Ausführung der Oberfläche.

An dieser Stelle seien folgende Variablen definiert:

$m :=$ Anzahl der Kostenobjekte (Zielgrößen)
 $n :=$ Anzahl der Iteration
 $o :=$ Anzahl der akzeptierten Iterationsschritte
 $p :=$ Anzahl der Stellgrößen

Eine Erläuterung der Variablen aus *daten.m* erfolgt in Tabelle 3.

Variable	Erklärung
Autor	String mit Namen des Autors
Datum	String mit Datum der Modellerstellung.
Algorithmus	String mit Namen des verwendeten Optimierungsalgorithmuses.
Modell	String mit Pfad und Namen des erstellten Modells.
jobs	Anzahl der Jobs.
OptAdjustObjectTable	$p \times 4$ Matlab-Cell. S1: Art der Stellgröße. S2: Spezifizierung der Stellgröße. S3: Auf welches Objekt angewendet. S4: Spezifizierung der Anwendung.
OptGetConfigurationDiag	$n \times p$ Matlab-Cell. $\text{OptGetConfigurationDiag}(i,j)$ entspricht dem Wert der j -ten Stellgröße bei der i -ten Iteration. Ist die Stellgröße j eine Permutation so ist $\text{OptGetConfigurationDiag}(i,j)$ eine $1 \times \text{jobs}$ Matlab-Cell, welche die jeweilige Permutation beschreibt.

3.1 Modelldaten

In Bereich **1** werden modelspezifischen Daten angezeigt. Der Autor entspricht dem Entwickler der Schnittstelle und Matlab-Oberfläche. Datum zeigt an, wann das Modell im sincron MODELLER erstellt wurde. Diese Information ist besonders nützlich beim Einladen eines älteren Modells (vgl. Abschnitt 3.9). Dessen Pfad und Name werden unter Punkt Modell angegeben. Standardmäßig erfolgt der gesamte Datenimport/export in dieses Verzeichnis (vgl. Abschnitt 3.7 und 3.9). Algorithmus spezifiziert den verwendeten heuristischen Optimierungsalgorithmus.

3.2 3D-Visualisierung des Optimierungsprozesses

In Segment **2** findet die Visualisierung des Optimierungsprozesses statt. Dabei werden je nach Achsenbelegung (vgl. Abschnitt 3.5) die Ziel- bzw. Stellgrößen jedes Iterationsschrittes im 3D-Diagramm als Kugel bzw. Kreuz (vgl. Abschnitt 3.6) dargestellt. Der Radius der Kugel variiert je nach Anzahl der Iterationsschritte. Das 3D-Diagramm lässt sich drehen, indem man die linke Maustaste auf dem Diagramm gedrückt hält und die Maus bewegt.

3.3 Ansichtsoptionen

Unter Punkt **3** ist es mit Hilfe der Azimuthrotation und Elevation möglich, den Blickwinkel auf das 3D-Diagramm genau einzustellen. Abbildung 7 erläutert diese beiden Winkel¹.

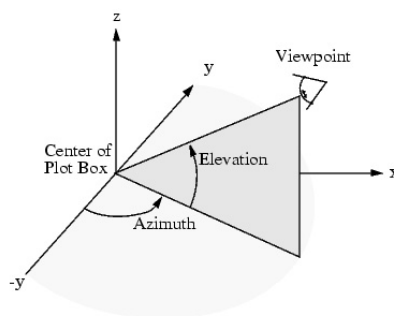


Abbildung 7: Erläuterung der Ansichtsoptionen

¹Bild entnommen aus der Matlab 7 Release 14 Hilfe.

Somit ergeben sich folgende geläufige Ansicht-Winkel-Kombinationen (Tabelle 4).

Ansicht	Azimuth	Elevation
x-z-Ebene	0°	0°
y-z-Ebene	90°	0°
x-y-Ebene	0°	90°

Tabelle 4: Ansicht-Winkel-Kombinationen

Die Aktualisierung des 3D-Diagrammes erfolgt bei Betätigung der Enter-Taste nach dem jeweiligen Winkeleintrag.

3.4 Korrelationskoeffizient

In Bereich 4 erfolgt die Berechnung des Korrelationskoeffizienten zwischen zwei visualisierten Größen. Im angezeigten Popup-Menü kann dabei die gewünschte Achsenkombination ausgewählt werden. Seien $X = (x_1, x_2, \dots, x_k)$ und $Y = (y_1, y_2, \dots, y_k)$ Wertepaare bezügl. der gewählten Achsenkombination. Dann berechnet sich der Korrelationskoeffizient aus:

$$\begin{aligned} \varrho(X, Y) &= \frac{Cov(X, Y)}{\sqrt{Var(X)} \cdot \sqrt{Var(Y)}} \\ &= \frac{\sum_{i=1}^k (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^k (x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^k (y_i - \bar{y})^2}} \end{aligned}$$

Dabei sind

$$\bar{x} = \frac{1}{k} \cdot \sum_{i=1}^k x_i \quad \text{und} \quad \bar{y} = \frac{1}{k} \cdot \sum_{i=1}^k y_i$$

die Erwartungswerte von X und Y .

Der Korrelationskoeffizient nimmt Werte zwischen -1 und 1 an. Dabei gilt:

- $\varrho = -1$: X und Y sind perfekt negativ korreliert, d.h. X und Y verhalten sich antiproportional zueinander.
- $\varrho = 0$: X und Y sind unkorreliert, d.h. es besteht kein statistischer Zusammenhang zwischen X und Y .
- $\varrho = 1$: X und Y sind perfekt positiv korreliert, d.h. X und Y verhalten sich proportional zueinander..

Nur ein signifikanter Korrelationskoeffizient berechtigt zur Aussage, es bestehe ein statistischer Zusammenhang zwischen X und Y . Als Hilfe wird zusätzlich die gewählte Achsansicht im Diagramm grün umrahmt.

3.5 Visualisierungsparameter

In Segment **5** ist es möglich die Achsenbelegung für das 3D-Diagramm festzulegen. Dabei werden Zielgrößen ohne Namen mit ihren Tcl-Bezeichnern (z.B. `cost1`, `cost2`, ... usw.) angezeigt. Stellgrößen werden durch Voranstellung von `SG_...` gekennzeichnet. Falls bei der Optimierung mit Permutationsstellgrößen gearbeitet wurde, werden diese mit ihrem jeweiligen Abstand zum gefundenen Optimum im Diagramm dargestellt. Das zur Brechnung verwendete Abstandsmaß ist in diesem Fall im Popup-Menü **Distanzmaß** wählbar.

3.6 Anzeigeoptionen

In Bereich **6** ist es möglich zwischen verschiedenen Anzeigevarianten zu wählen bzw. diese miteinander zu kombinieren. Grundsätzlich muss man sich, wie bereits in Abschnitt 3.2 angedeutet, zwischen zwei Visualisierungsvarianten entscheiden. Wurde **Kreuze (Einfarbig)** aktiviert, werden alle Iterationsschritte als Kreuz im 3D-Diagramm angezeigt. Deren Farbe ist im rechtsstehenden Popuptmenu auswählbar. Wurde **Kugeln (Einfarbig / Colormap)** aktiviert, werden alle Iterationsschritte entweder als einfarbige bzw. bei zusätzlicher Aktivierung von **Colormap** als verschiedenfarbige Kugeln angezeigt. Nach der Auswahl von **Colormap** erfolgt automatisch ein Update des rechtsstehenden Popuptmenus in dem nun nicht mehr einzelne Farben sondern Farbschemas auswählbar sind. Hierbei wird die Iterationsreihenfolge berücksichtigt. Deren zeitlicher Ablauf wird in einer Vorschau unter dem Popuptmenu angezeigt (vgl. z.B. Abbildung 13 auf Seite 18). Man beachte, dass

sich die beiden bisher angesprochenen Visualisierungsoptionen gegeneinander ausschließen. Abbildung 8 zeigt jeweils ein Beispiel für diese.

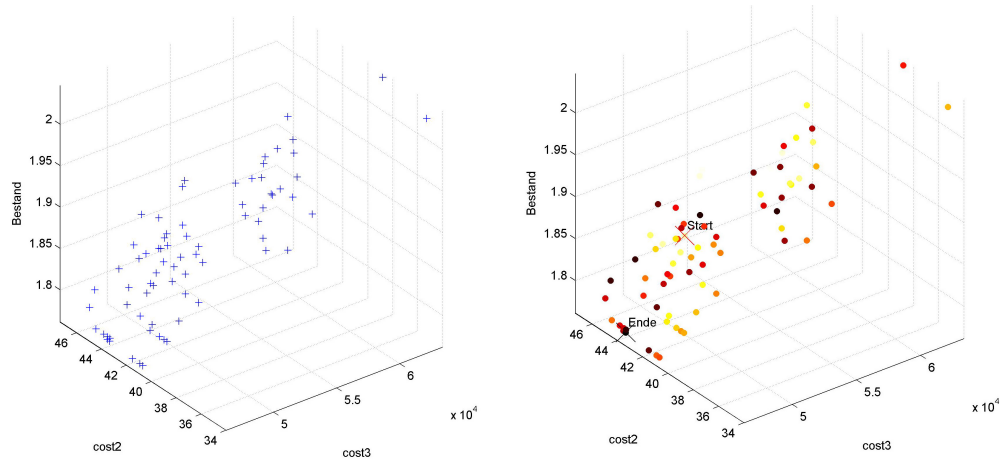


Abbildung 8: Visualisierungsmöglichkeiten

Desweiteren existieren noch acht weitere Checkboxes, welche optional und unabhängig voneinander aktiviert werden können. Wählt man **Linie gem. zeitlichen Ablauf**, so werden die aufeinanderfolgenden Iterationsergebnisse durch eine blaue gestrichelte Linie miteinander verbunden. Wie man jedoch in Abbildung 9 erkennen kann, ist diese Option nur sinnvoll, wenn die Anzahl der Iterationen eher gering ist.

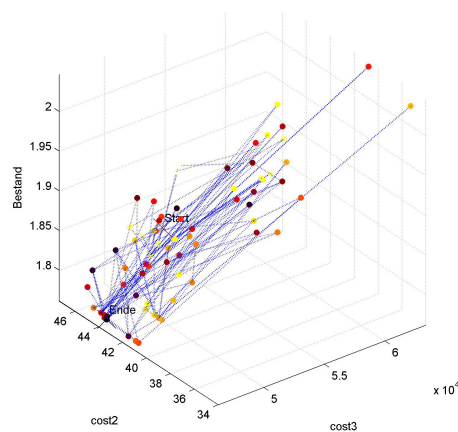


Abbildung 9: Option: Linie gem. zeitlichen Ablauf

Als günstiger erweist es sich in diesen Fällen mit der Option **Linie akzeptierte Punkte** zu arbeiten. Diese verbindet nur die vom Optimierer akzeptierten Iterationsschritte miteinander. Je nach Art des verwendeten Optimierers können dies z.B. alle Iterationsschritte, welche eine Verbesserung des Zielfunktionswertes zur Folge haben, sein. Diese Linie wird rot im 3D-Diagramm dargestellt (vgl. Abbildung 10).

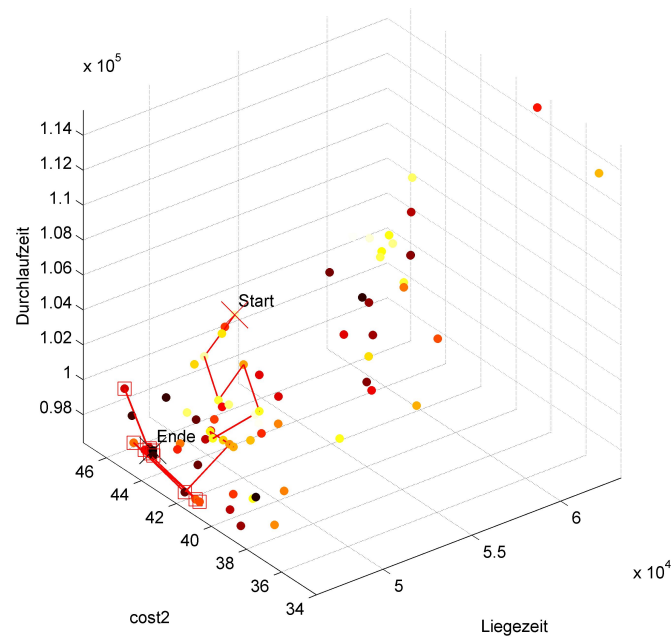


Abbildung 10: Checkbox 4 und 6

Desweiteren ist es möglich mit Hilfe der Checkbox **Doppelte Stellgrößenkombination** jene Iterationsschritte anzuzeigen, welche mehrfach berechnet wurden, was zum Beispiel Rückschlüsse auf die Effektivität des verwendeten Optimierungsalgorithmusses zulässt. Die Kennzeichnung erfolgt als grüner Kreis um den jeweiligen Datenpunkt.

Eine weitere wählbare Option bietet die Checkbox **Die besten ...%**. Hier werden die besten, in Prozent anzugebenden, Optimierungsergebnisse mit einem roten Rahmen hervorgehoben (vgl. Abbildung 10). Existieren Iterationsergebnisse, deren Zielfunktionswert mit einem der hervorgehobenen identisch ist, so werden diese ebenfalls rot umrahmt.

Mit Hilfe der Checkbox **Pareto-Front** ist es möglich eine Pareto-Front zwischen zwei visualisierten Zielgrößen anzeigen zu lassen. Die Auswahl der Zielgrößen kann dabei unter **Paretooptionen** verändert werden. Die Pareto-Front verbindet alle nichtdominierten Punkte bzgl. der gewählten Zielgrößen und ist somit in der 2D Ansicht als Linie (blau gestrichelt) bzw. in der 3D-Ansicht als Fläche im Diagramm zu sehen (vgl. Abb. 11).

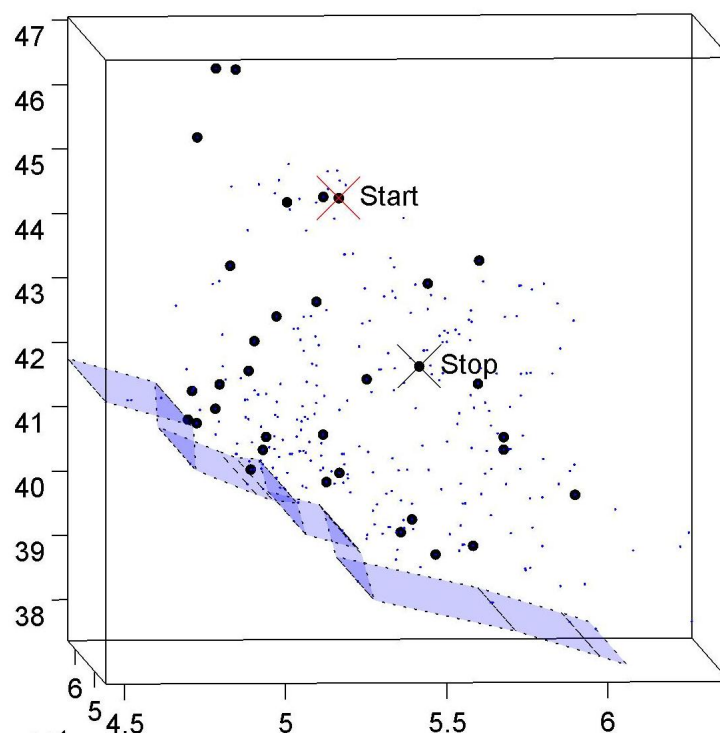


Abbildung 11: Checkbox 7 und 8

Mit Hilfe des Buttons **Hintergrund speichern** ist es möglich eine Punktwolke modellgebunden als Hintergrund zu hinterlegen. Optimiert man das Modell ein weiteres mal, ist es dann möglich diesen Hintergrund einzublenden (Checkbox **Hintergrund anzeigen** aktivieren). Dieser wird dabei als feine blaue Wolke angezeigt (vgl. Abb. 11).

Durch Betätigung des Buttons **Anzeigen** findet eine Aktualisierung des in Abschnitt 3.2 beschriebenen 3D-Diagramms statt. Dabei ist zu beachten, dass bei Modellen mit hoher Iterationszahl (> 3000) die Option **Colormap**

eine spürbare Verlängerung der Rechendauer zur Folge hat. Es wird daher empfohlen dann nur mit einer einfarbigen Anzeige zu arbeiten. Start und Ende der Berechnung werden durch eine entsprechende Meldung im Statusfenster (vgl. Abschnitt 3.8) angezeigt (vgl. Abbildung 13). Die letzten beiden Anzeigeeoptionen bilden die Checkboxes **Box** und **Grid**. Mit ihnen ist es möglich eine Box um das 3D-Diagramm bzw. das Grid im 3D-Diagramm zu aktivieren/deaktivieren. Die Abbildungen 10 bzw. 11 zeigen die Unterschiede.

3.7 Bildexport

In Segment 7 ist es möglich das dargestellte 3D-Diagramm benutzerfreundlich als Grafik zu exportieren. Diese Funktion wurde implementiert, da es des öfteren Probleme mit der herkömmlichen Matlab-Bildexport-Funktion gab. Um eine Grafik zu exportieren muss man zuerst den Button **Vollbild** betätigen. Es öffnet sich das 3D-Diagramm in einer neuen Figure (Vollbildmodus). In dieser Figure stehen die bekannten Matlab-Bildbearbeitungsprogramme, wie zum Beispiel: "Plot Edit Toolbar", "Plot Browser" oder "Property Editor" zur Verfügung. So ist es zum Beispiel möglich der Grafik einen Titel zu geben (vgl. Abbildung 12).

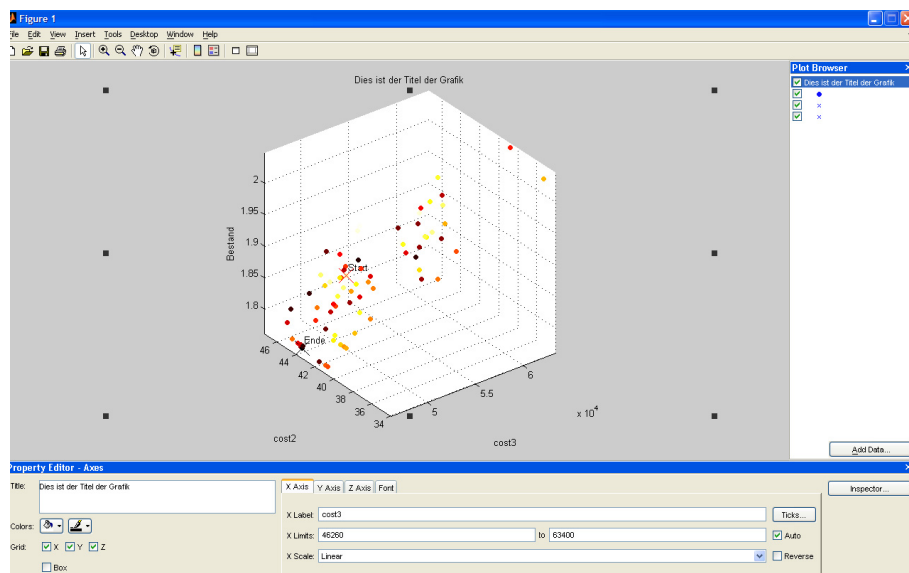


Abbildung 12: Plot-Optionen

Nachdem die Grafik im Vollbildmodus nach dem Willen des Nutzers bearbei-

tet wurde, ist der Vollbildmodus durch **Minimierung** des Fensters zu beenden (Fenster nicht schließen!). Im Segment 8 des GUIs ist ein Durchsuchen-Button zu sehen (vgl. Abbildung 13). Wie bereits in Abschnitt 3.1 angedeutet wird standartmäßig der Pfad und Name des Modells für das zu erstellende Bild gesetzt. Eine Änderung dessen ist jedoch mit Hilfe des Durchsuchen-Buttons möglich. Man beachte dabei, dass keine Dateierweiterung gesetzt werden darf. Diese ist unter **Bildtyp** zu wählen. Unter **Auflösung dpi** ist die gewünschte Auflösung

$$0 < dpi \leq 600$$

anzugeben. Beim Versuch ein Bild ohne geöffnetes Vollbild zu speichern, wird eine entsprechende Fehlermeldung in der Statuszeile herbeigeführt.

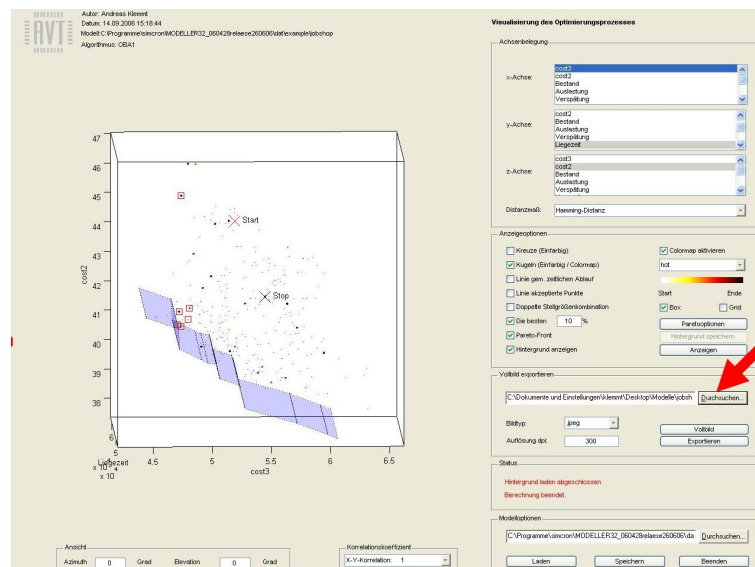


Abbildung 13: Durchsuchen-Button

Wird eine Auflösung außerhalb der oben angegebenen Grenzen eingegeben, hat dies ebenfalls eine Fehlermeldung zur Folge. Die Abbildungen 8, 9 oder 10 sind Beispiele für die eben erläuterte Exportfunktion.

3.8 Statusfenster

In Sement 8 befindet sich das Statusfenster. Wie bereits mehrfach angedeutet, werden alle Meldungen, Warnungen und Hinweise in diesem Bereich in roter Schrift angezeigt. In Tablle 5 sind alle möglichen Ereignisse erläutert.

Ereignis	Bedeutung
Kein Bildexport. Vollbildmodus nicht aktiviert.	Der Vollbildmodus wurde noch nicht aktiviert bzw. es wurde Exportieren betätigt nachdem das Vollbild wieder geschlossen wurde.
Bitte $0 < dpi \leq 600$ wählen.	Es wurde versucht ein Bild mit einer Auflösung $0 < dpi \leq 600$ zu exportieren.
Berechnung läuft.	Dieses Ereignis wird während der Berechnung des 3D-Diagramms angezeigt. Alle GUI Funktionen sind nicht verfügbar. Es nicht möglich das GUI zu schließen.
Hinweis: Vorgang kann mehrere Minuten dauern. Berechnung läuft.	Zusätzlich zum eben genannten Fall sind hier Art und Umfang der Berechnung sehr zeitintensiv. Der Nutzer wird mit dieser Meldung explizit darauf hingewiesen sich zu gedulden.
Berechnung beendet.	Die Berechnung ist beendet und somit alle GUI Funktionen wieder verfügbar.
Prozentsatz >0 und <100 wählen.	Falsche Eingabe bei der Visualisierung der besten Punkte.
Hintergrund wird geladen. Einen Moment bitte.	Hintergrundbild ist vorhanden und wird eingeladen
Hintergrund laden abgeschlossen.	Die Generierung der Hintergrundwolke ist abgeschlossen.

Kein Hintergrund für dieses Modell vorhanden.	es wurde kein modellspezifischer Hinter- grund gefunden.
Hintergrund gespeichert.	Es wurde ein modellspezifischer Hinter- grund erzeugt.

Tabelle 5: Ereignisse Statusfeld

3.9 Modell-Export/Import

In Bereich **9** ist das Laden bzw. Speichern von Modellen möglich. Will man ein Modell für eine spätere Verwendung abspeichern, so ist mit Hilfe des Durchsuchen-Buttons ein entsprechendes Verzeichnis zu wählen und ein Modellname anzugeben. Dabei ist zu beachten, dass die Dateierweiterung `.mat` automatisch ergänzt wird. Das Betätigen des Speicher-Buttons schließt den Vorgang ab. Beim Laden eines bereits vorhandenen Modells geht man analog vor. Man wählt dieses mit Hilfe des Durchsuchen-Buttons aus (mit Dateierweiterung) und betätigt den Laden-Button.

Für weitere programmtechnische Details sei an dieser Stelle wieder auf den Quellcode verwiesen.