OP-Scheduling – Erweiterung

Bericht der Erweiterung des OP-Scheduling

Fach: Angewandte Betrieblich Systemforschung

Professor: Prof. Dr. Michael Grütz

Autor: Andreas Bahr Julia Pawaletz

Mat-Nr.: 287891 288002

E-Mail: [Andreas.Bahr@htwg-konstanz.de](mailto:Andreas.Bahr@htwg-konstanz.de), [Julia.Pawaletz@htwg-konstanz.de](mailto:Julia.Pawaletz@htwg-konstanz.de)

Semester: 7. Semester Wirtschaftsinformatik (WS15/16)

Datum: 20.01.2016

Inhaltsverzeichnis

[1 Verwendung des OP Scheduling 3](#_Toc441653619)

[1.1 Schritt für Schritt Anleitung 3](#_Toc441653620)

[1.1.1 Öffnen des OP Scheduling 3](#_Toc441653621)

[1.1.2 Erstellen einer neuen Datei 4](#_Toc441653622)

[1.1.3 Voreinstellungen 4](#_Toc441653623)

[1.1.4 Daten in Tabelle eintragen und optimieren 4](#_Toc441653624)

[1.1.5 Ausgabe der Lösung 5](#_Toc441653625)

[1.1.6 Speichern 5](#_Toc441653626)

[1.1.7 Einbinden von Solvern 5](#_Toc441653627)

[2 LP – Ansatz 6](#_Toc441653628)

[2.1 Anwendungsbeispiel 6](#_Toc441653629)

[2.2 LP – Modell 6](#_Toc441653630)

[2.3 Beschreibung der Restriktionen / Interpretation 7](#_Toc441653631)

[3 Grenzwertbasierte Tests 8](#_Toc441653632)

[4 Stand vor der Überarbeitung 9](#_Toc441653633)

[4.1 Features Realisiert 9](#_Toc441653634)

[4.2 Features noch nicht Realisiert 9](#_Toc441653635)

[5 Überarbeitung 10](#_Toc441653636)

[5.1 Solver-Pfad Anpassung 10](#_Toc441653637)

[5.2 Erweiterung auf bis zu 8 OP Säle 10](#_Toc441653638)

[5.3 Features Realisiert 10](#_Toc441653639)

[5.4 Features noch nicht Realisiert 11](#_Toc441653640)

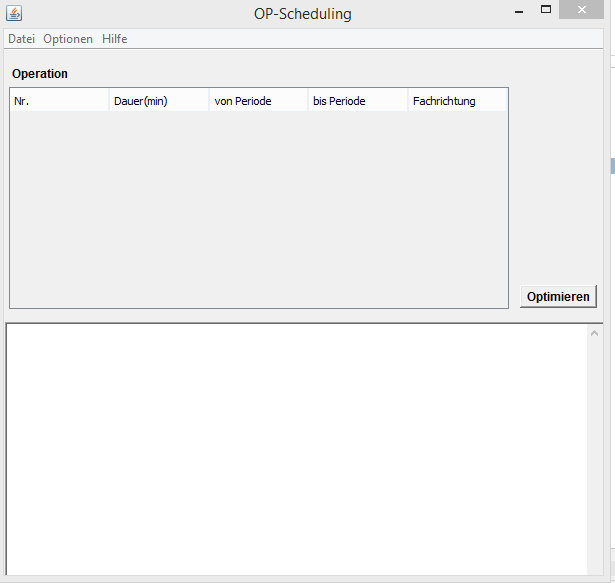
# Verwendung des OP Scheduling

Der OP Scheduling sorgt für eine bestmögliche Auslastung der OP – Säle und automatisiert die Zuordnung der verschiedenen OPs. Für die Auslastungsoptimierung wird die Maximalbelegungsdauer eines Saales gering gehalten. Dabei wird versucht gleich hohe oder geringere Belegungsdauern für die anderen Säle zu erreichen.

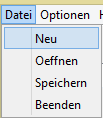
## Schritt für Schritt Anleitung

Im Folgenden wird gezeigt, wie der OP Scheduling genutzt wird. Zu finden ist die Methode in OR Alpha unter der Kategorie A.

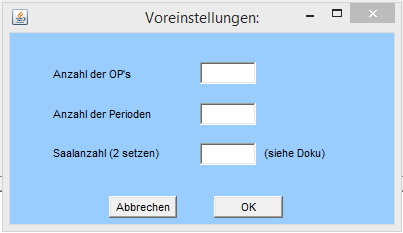
### Öffnen des OP Scheduling



### Erstellen einer neuen Datei

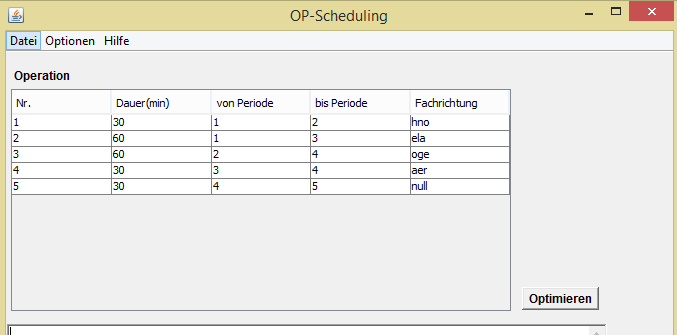


### Voreinstellungen



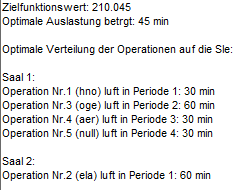
In diesem Fenster müssen die Voreinstellungen getroffen werden. Hierzu werden die Anzahl der Operationen, die Anzahl der Perioden (wobei eine Periode 60 Minuten dauert) und die Saalanzahl festgelegt.

### Daten in Tabelle eintragen und optimieren



In die automatisch erstellte Tabelle werden nun die Operationen mit ihrer Dauer und Fachrichtung eingetragen. Bei der Dauer ist zu beachten, dass sie in Minuten angegeben wird. Nach dem Eintragen wird der Button „Optimieren“ geklickt.

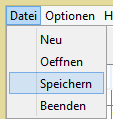
### Ausgabe der Lösung



Die Lösung wird im unteren Teil des Fensters angezeigt. Aus der Lösung sind der Zielfunktionswert und die Optimale Ausleistung ersichtlich. Im Weiteren wird genau dargestellt, welche Operation in welchem Raum und in welcher Periode durchgeführt wird.

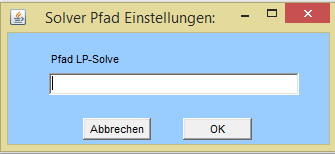
### Speichern

Das erstellte Problem kann gespeichert und zu einem späteren Zeitpunkt wieder geöffnet werden:



### Einbinden von Solvern

Über Optionen 🡪 Solver Pfad



Hier kann der Solver Pfad manuell vom Benutzer angepasst und geändert werden.

# LP – Ansatz

Der LP – Ansatz wird anhand eines Anwendungsbeispiels aufgezeigt. Eine Periode geht immer genau 60 Minuten.

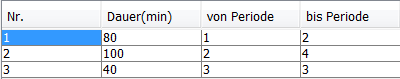
## Anwendungsbeispiel

Tagesablauf (Schicht 1: 8 – 13 Uhr):

* + 100 min zwischen 8 und 10 Uhr (Maier)
  + 70 min zwischen 8 und 9 Uhr (Schilling)
  + 90 min zwischen 9 und 11 Uhr (Müller)
  + 45 min zwischen 9 und 10 Uhr (Stoll)
  + 160 min zwischen 11 und 13 Uhr (Kaiser)

Zwei Säle stehen bei der Planung zur Verfügung.

## LP – Modell



**Variable:**

Y = Auslastung

Xij = Anzahl der Stunden, für die Operation i in der Periode j

Si = Schaltvariable (ZERO/ONE-Variable)

**Zielfunktion**: Z = Y 🡪 min!

**Restriktionen**:

* NB1: X11 + X12 = 80
* NB2: X22 + X23 + X24 = 100
* NB3: X33 = 40
* NB4: X11 - 1Y <= 0
* NB5: X12 + X22 - 2Y <= 0
* NB6: X23 + X33 - 2Y <= 0
* NB7: X24 - 1Y <= 0
* NB8: X11 <= 60
* NB9: X12 + X22 <= 120
* NB10: X23 + X33 <= 120
* NB11: X24 <= 60
* NB12: Nichtnegativitätsbedingung 🡪 Y >= 0, Xij >= 0

## Beschreibung der Restriktionen / Interpretation

Die Nebenbedingungen 1 bis 3 fassen die jeweiligen Operationen innerhalb der/den Periode(n), in der/den sie ausgeführt werden sollen zusammen. Die Periodenwerte müssen die rechte Seite genau erfüllen (Gleichheitsbeziehung).

Die Nebenbedingungen 4 bis 7 fassen die Aufteilung der verschiedenen Operationen auf die jeweiligen Perioden zusammen unter Berücksichtigung der Anzahl der Säle je Periode.

Die Nebenbedingungen 8 bis 11 fassen die Aufteilung der verschiedenen Operationen auf die jeweiligen Perioden zusammen unter Berücksichtigung der maximalen Kapazität der eingeplanten Sälen. Die zur Verfügung stehende Periodendauer beträgt 60 min.

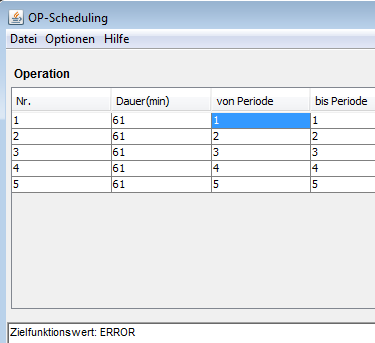
Zielfunktion:

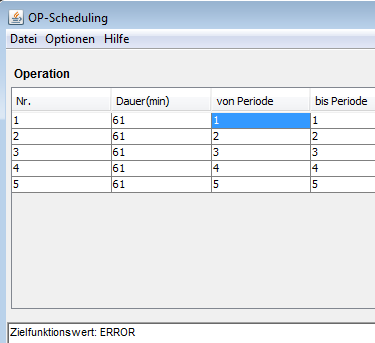
Z = Y -> min!

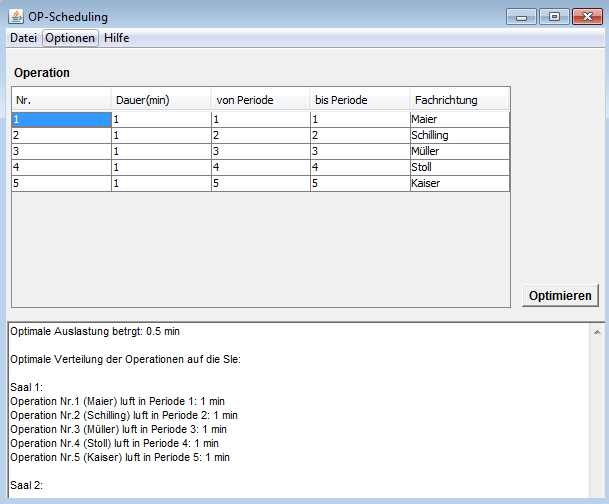
Durch die Restriktionen 4 bis 7 und Minimierung der Zielfunktion Z = Y wird versucht, eine möglichst niedrige und somit gleichmäßige Auslastung zu erreichen (Glättung des Kapazitätsprofils über die Perioden).

# Grenzwertbasierte Tests

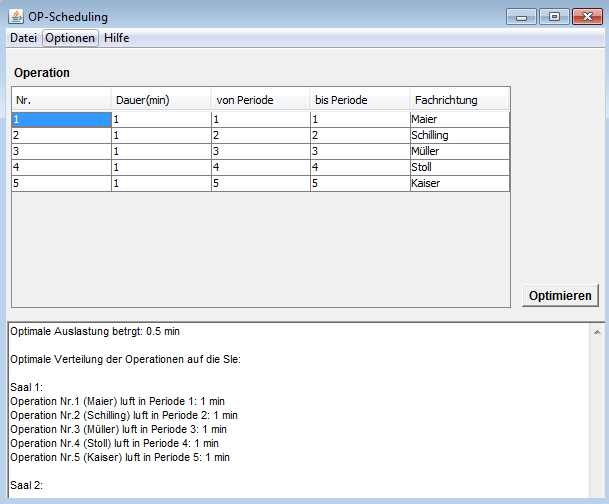
* Nur Ganzzahligkeit
* „0“ erlaubt
* Obere Grenze

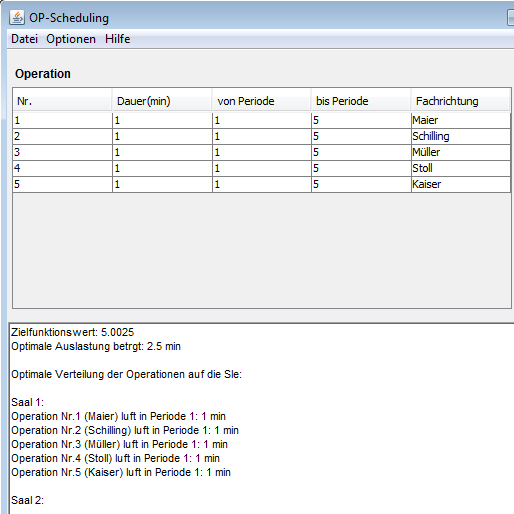




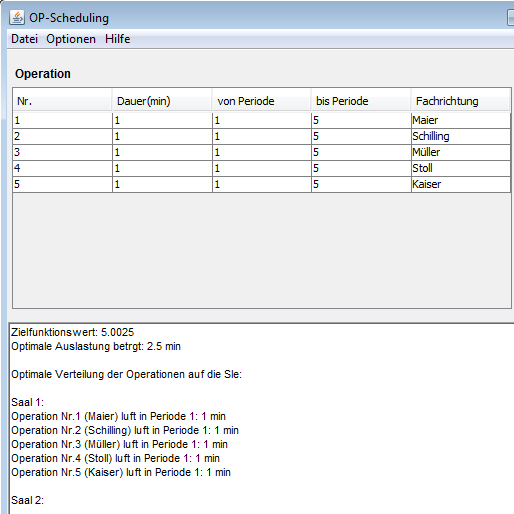


* Untere Grenze
* „minus“ wird behandelt wie „0“

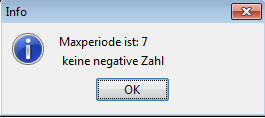




* Offene Perioden
* Geschlossene Perioden



* Flasche Periode (negativ/0)



# Stand vor der Überarbeitung

In diesem Kapitel wird der Stand vor der Überarbeitung aufgelistet.

## Features Realisiert

Vor der Überarbeitung waren folgende Features Realisiert:

* Daten können eingegeben werden
  + Über das GUI können alle nötigen Daten eingegeben werden.
  + Falsche Eingaben werden abgefangen
* LP-Ansatz wird erzeugt
  + Durch bestätigen der Eingaben wird der LP-Ansatz erzeugt
* Speichern/Öffnen funktioniert
  + Das Speichern aller eingegebenen Daten ist möglich und funktioniert
  + Es wird die Dateiendung \*.opsa genutzt
  + Es ist ein Filter eingebaut, damit nur Dateien des mit der Endung \*.opsa angezeigt werden
  + Die gespeicherten Daten können geöffnet werden
  + Hier wurde der selbe Filter wie für die Speichern-Funktion genutzt
* Der Solver Pfad kann geändert werden und wird in einer INI-Datei gespeichert
  + Der Solver Pfad kann über das GUI eingegeben werden
  + Es wird überprüft ob die Datei welche angegeben wird auch vorhanden ist

Diese Features haben ohne größere Fehler funktioniert, wobei der LP-Ansatz von uns nicht überprüft wurde.

## Features noch nicht Realisiert

* Es kann keine Wartung für einen Saal angegeben werden
  + Es ist nicht möglich eine Wartung für einen Saal zu planen
* **Es wird kein Ergebnis ausgegeben**
  + **Nach dem Drücken des Button „Optimieren“ wird kein Ergebnis ausgegeben**
* Anzahl der Operationssäle ist auf 2 beschränkt
  + Es ist nicht möglich mehr oder weniger als 2 Operationssäle auszuwählen
* Grafischer Belegungsplan wird nicht ausgegeben
  + Es ist nicht möglich das Ergebnis Grafisch einzusehen
* Die Belegung von möglichst wenig Sälen kann nicht berechnet werden
  + Es kann nur die optimale Auslastung berechnet werden.

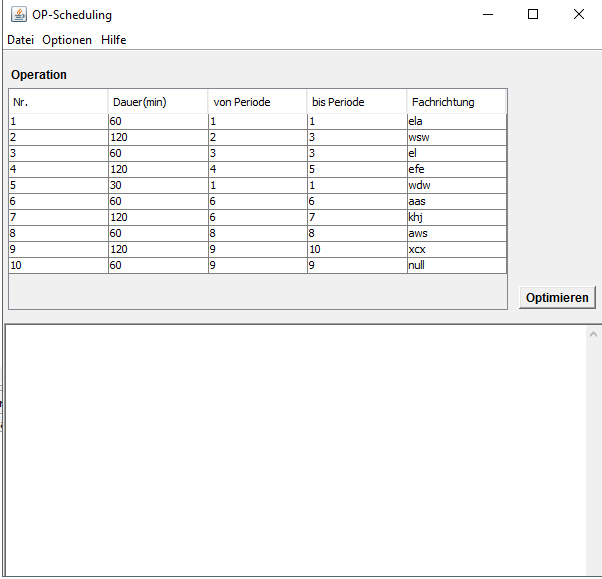
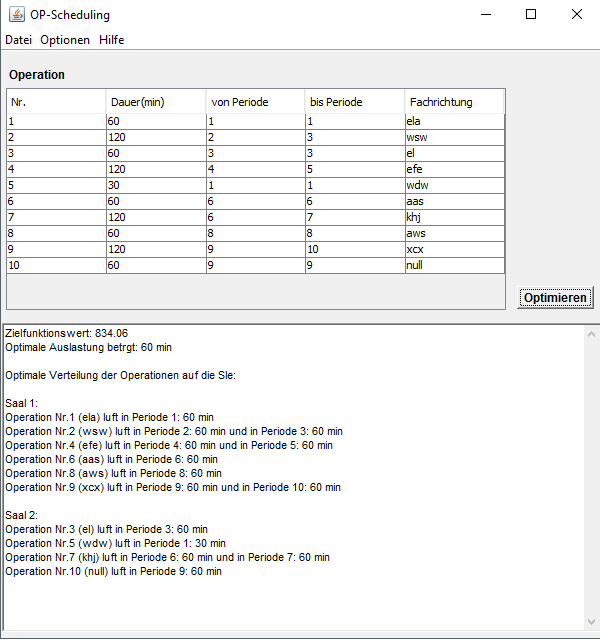
Diese Features waren nicht Implementiert.

# Überarbeitung

In dem Projekt soll der OP Scheduling wieder Ergebnisse ausgeben und nicht nur genau mit zwei OP – Sälen, sondern mit bis zu 8 Sälen rechnen können.

## Solver-Pfad Anpassung

Nach einer Analyse der gegebenen Methode, konnte festgestellt werden, dass eingegebenen Daten in der Oberfläche nicht an LP-Solve übergeben werden. Dadurch war keine Ausgabe der Lösung möglich. Zu Anfangs bestand die Vermutung, dass es an der Pfad-Einstellung in der ini.java liegen könnte. Durch ein Meeting, mit dem Projektverantwortlichen Serkan Önnisan, wurde uns geraten mit der Solver-Klasse(iniPaths.java) aus dem Ernährungsplaner zu arbeiten. Daraus folgte eine funktionierendes Programm und eine erfolgreiche Ausgabe. Nach weiteren Testphasen, stellten die Projektmitglieder, ein Problem in der Solver-Pfad Änderung fest. Das Problem beinhaltete eine Änderung der Pfade, welche nicht in die ini.txt übergeben wurde. Daraus folgte die Entscheidung zurück zur Ursprungsklasse zu gelangen, welche eine erfolgreiche Pfade-Änderung zulässt. Aus der Erfahrung mit der Einbindung der iniPaths.java, aus dem Programm Ernährungsplaner, gelang es dem Team das Programm mit alter Struktur wieder lauffähig und ausgabefähig zu gestalten.

## Erweiterung auf bis zu 8 OP Säle

Die Problematik hierbei ist, dass der Programmcode des OP Scheduling mit Matrizen programmiert ist. Die Matrizen zu erweitern, verzerrt das Ergebnis des LP-Solves und stößt ihn an seine Grenzen. Mittels Switch-Case Verzweigungen, wurden die Matrizen (OP-Säle) erweitert. Diese Erweiterung führt dazu, dass der OP-Scheduling in der Konsole die Variablen Anzahl erhöht und teilweise richtig berechnet. Aufgrund stückweiser falscher Berechnungen wurde dieses Unterfangen belassen und die Anzahl der OP Säle auf zwei festgelegt, dass es zu keiner fehlerhaften Ausgabe führen kann.

## Features Realisiert

* Der OP – Scheduling gibt wieder eine richtige Lösung aus.
* Struktur zur Erweiterung der OP Saal Anzahl erstellt.
* Auslesen und Überschreiben der Solver-Pfade aus einer INI-Datei.

## Features noch nicht Realisiert

* Es kann keine Wartung für einen Saal angegeben werden
  + Es ist nicht möglich eine Wartung für einen Saal zu planen
* Grafischer Belegungsplan wird nicht ausgegeben
  + Es ist nicht möglich das Ergebnis Grafisch einzusehen
* Die Belegung von möglichst wenig Sälen kann nicht berechnet werden
  + Es kann nur die optimale Auslastung berechnet werden.

Diese Features sind noch nicht Implementiert.