Optimierung der Blister-Verkäufe

Um die Bestellungen automatisch zu verkaufen, werden jeweils die ersten Eber einer Bestellung berücksichtigt und die Blister möglichst optimal auf die Bestellungen verteilt.

Die Lösung besteht aus der Anwendung einer linearen Optimierung.

**Definitionen**

Nach Aussen sichtbare Variablen:

|  |  |  |
| --- | --- | --- |
| Variable | Zulässiger Bereich | Beschreibung |
|  |  | Status der Bestellung n (1 = aktiv, 0 = inaktiv) |
|  |  | Langzeitanteil der Bestellung n |

Intern abgeleitete Variablen:

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Zulässiger Bereich | Ableitung | Beschreibung |
|  |  |  | Menge der neuen Blister |
|  |  |  | Menge der alten Blister |
|  |  |  | Reduktionsmenge der alten Blister |

Konstanten

|  |  |  |
| --- | --- | --- |
| Konstante | Zulässiger Bereich | Beschreibung |
|  |  | Gewicht der Erfüllung der Bestellung n |
|  |  | Status der Gewichtung der Höhe des Langzeitanteils |
|  |  | Status der Gewichtung der Höhe des Kurzzeitanteils |
|  |  | Faktor zur Gewichtsbestimmung |

Parameter

|  |  |  |
| --- | --- | --- |
| Parameter | Zulässiger Bereich | Beschreibung |
|  |  | Gesamtbestellmenge der Bestellung n |
|  |  | Gesamte verfügbare Anzahl neuer Blister |
|  |  | Gesamte verfügbare Anzahl alter Blister |
|  |  | Mindestlangzeitanteil (kann gesetzt werden oder nicht). |

Anhand der definierten Variablen und Zustände lässt sich die Zielfunktion folgendermassen beschreiben:

Die erste Summe der Zielfunktion bezieht alle zu erfüllenden Bestellungen ein. Je mehr Bestellungen erfüllt werden können, desto besser. Da dies das wichtigste Kriterium ist, wird das Gewicht auf 10000 gesetzt. Die einzelnen Gewichte (WEi) werden nach Bestelldatum aufsteigend weniger hoch sein (die grösste Priorität hat die erste erfasste Bestellung, dann die Zweite, usw.).

Die zweite Summe der Zielfunktion bezieht alle Langzeitanteile einer Bestellung ein. Die Gewichtung ist pro Bestellung unterschiedlich. Das Gewicht WLi variiert zwischen 0 und 1 (ist der Wert 0, so wird die Grösse des Langzeitanteils nicht berücksichtigt). Um die Verteilung gerechter zu gestalten (die frühesten Bestellungen sollen am meisten der neuen Blister bekommen), werden zwei weitere Faktoren, die später beschrieben werden, berücksichtigt.

Die dritte Summe der Zielfunktion bezieht alle Kurzzeitanteile einer Bestellung ein. Die Gewichtung ist pro Eintrag unterschiedlich. Das Gewicht WSi variiert zwischen 0 und 1 (ist der Wert 0, so wird die Grösse des Kurzzeitanteils nicht berücksichtigt). Um die Verteilung gerechter zu gestalten (die frühesten Bestellungen sollen am meisten der neuen Blister bekommen), werden zwei weitere Faktoren, die später beschrieben werden, berücksichtigt.

Die Gewichte WLn und WSn einer Bestellung n sind zusammenhängend. Eines von beiden ist immer 0, das Andere 1.

Wenn ein Mindestlangzeitanteil Ln gesetzt wurde, so soll nach Kurzzeitanteil optimiert werden, also möglichst alle alten Blister verkaufen. Wenn kein Mindestlangzeitanteil Ln gesetzt wurde, so soll nach Langzeitanteil optimiert werden, also möglichst alle neuen Blister verkaufen.

Im Folgenden werden weitere Nebenbedingungen beschrieben, die bei der Optimierung berücksichtigt werden müssen.



Wenn en = 1 ist:

1. Wenn kein Mindestlangzeitanteil gesetzt wurde, so gilt a, ansonsten b.

Wenn en = 0 ist:



In Worten:

1. Alle verkauften neuen Blister übersteigen die verfügbare Menge nicht.
2. Alle verkauften alten Blister übersteigen die verfügbare Menge nicht.
3. Die alten sowie die neuen Blister ergeben die Bestellmenge der Bestellung n.
4. Die Menge neuer Blister der Bestellung n ergibt sich durch die Multiplikation des Faktors Langzeitanteil mit der Bestellmenge der Bestellung n.
5. Wenn die Bestellung verarbeitet werden kann (en = 1), so darf der Langzeitanteil der Bestellung n zwischen 0 und 1 variieren (0 = nur alte Blister verkaufen, 1 = nur neue Blister verkaufen). Die Werte sind als Prozentangaben zu verstehen. Jedoch darf der Mindestlangzeitanteil Ln nicht unterschritten werden, wenn dieser gegeben ist.
6. Wenn die Bestellung verarbeitet werden kann, so ist die Reduktionsmenge der alten Blister gleich der Menge der zu verkaufenden alten Blister.
7. Wenn die Bestellung nicht verarbeitet werden kann (en = 0), so ist der Langzeitanteil der Bestellung n gleich 0.
8. Wenn die Bestellung nicht verarbeitet werden kann, so ist die Reduktionsmenge der alten Blister gleich 0.

# Zusammenhang zwischen Reduktionsmenge (rn) und Menge alter Blister (an)

Wenn eine Bestellung ausgeschaltet wird, müssen weiterhin alle Nebenbedingungen erfüllt sein. Da im Falle einer Bestellung, die nicht erfüllt werden kann (en = 0), der Langzeitanteil gleich 0 ist und dementsprechend auch die Menge neuer Blister gleich 0 ist, ist die Menge alter Blister durch Nebenbedingung III gleich der gesamten Bestellmenge der Bestellung n (Bn). Diese beiden Mengen (nn und an) werden zur Überprüfung der Nebenbedingungen I und II verwendet. Für die Nebenbedingung I müssen keine Anpassungen vorgenommen werden, da die Menge neuer Blister gleich 0 ist (für die Bestellung n werden also keine neuen Blister vergeben, was demnach auch zu keinem Überlauf der vorhandenen Menge neuer Blister führen wird).

Würde die Nebenbedingung II nun ebenfalls über die Anzahl der alten Blister der Bestellung n definiert, so könnte nie eine gültige Lösung gefunden werden, da die Menge der alten Blister womöglich immer überstiegen wird. Eine Bestellung kann also nicht ausgeschaltet werden.

Deswegen wird eine neue Variable (rn) eingeführt, welche die Reduktionsmenge der alten Blister beschreibt. Die Nebenbedingung II lässt sich also folgendermassen formulieren:

Wobei für rn gilt:

Die Multiplikation einer binären Variablen (en) und einer freien Variablen (an) lässt sich über vier Ungleichungen beschreiben.

Daraus folgt für die Umsetzung im Code:

Wobei M eine genug grosse Zahl ist (mindestens an). Im Code wird M auf 1'000'000 gesetzt (So viele alte Blister werden kaum in einer Bestellung verwendet).

Durch diese Variante wird die Reduktionsmenge der alten Blister in Nebenbedingung II im Fall, wenn en gleich 0 ist, ebenfalls auf 0 gesetzt. So können alle Nebenbedingung erfüllt werden, wenn eine Bestellung als inaktiv markiert wird.

# Genaue Erläuterung der Gewichte

Die erste Summe der Zielfunktion bezieht sich auf die zu erfüllenden Bestellungen. Diese sollen möglichst maximal sein, weswegen das Gewicht () hoch gewählt wird. Des Weiteren ist wichtig, dass die Bestellung dem Eingang nach abgearbeitet werden. Ältere Bestellungen haben also ein höheres Gewicht als Neue. Diese Gewichtung wird über WEn ausgedrückt.

Um den gewünschten Effekt möglichst häufig zu erzielen, müssen die ersten Bestellungen einen hohen Faktor haben. Dazu wurde die Funktion gewählt, wobei x dem Index der Bestellung n entspricht. Ab der 6. Bestellung werden diese gleich gewertet.

Das Mindestgewicht ist also. Das bedeutet, der Score, welchen es für die Erfüllung einer der niedrigsten gewichteten Bestellung gibt, beläuft sich auf 156.25.

Für die weiteren Gewichte ist die Definition des Parameters D wichtig.

Wobei a die Anzahl der Bestellungen ist.

Der Index läuft von 0 nach (n-1), wobei die ältesten Bestellungen zuerst abgearbeitet werden, also einen niedrigen Index besitzen.

Die zweite Summe der Zielfunktion bezieht sich auf die Gewichtung eines hohen Langzeitanteils.

Wenn kein Mindestlangzeitanteil angegeben wurde, so soll der Langzeitanteil möglichst maximiert werden. Das Gewicht WLn wird also auf 1 gesetzt. Nun sollen der Fairness geschuldet die ältesten Bestellungen möglichst viele neue Blister erhalten. Die Variable nn soll für alte Bestellungen n also möglichst maximiert werden. So wird der Faktor zu ln multipliziert.

Zur Veranschaulichung wird ein Beispiel angeführt:

|  |  |  |  |
| --- | --- | --- | --- |
| Bestellung |  |  |  |
| A | 1 | 3 |  |
| B | 2 | 2 |  |
| C | 3 | 1 |  |

Der Anreiz der Bestellung A die meisten neuen Blister zu geben, ist so um einiges höher als der Bestellung C.

Die dritte Summe der Zielfunktion bezieht sich auf die Gewichtung eines hohen Kurzzeitanteils.

Wenn ein Mindestlangzeitanteil angegeben wurde, so soll der Kurzzeitanteil möglichst maximiert werden. Das Gewicht WSn wird also auf 1 gesetzt.

Der Kurzzeitanteil der Bestellung n lässt sich durch den Langzeitanteil beschreiben:

Nun sollen wie beim Langzeitanteil die ältesten Bestellungen möglichst viele neue Blister erhalten. Die Variable nn wird für die ersten Bestellungen also möglichst maximal gehalten unter der Berücksichtigung, dass möglichst alle alten Blister verkauft werden. So wird der Faktor zu ln multipliziert.

Zur Veranschaulichung wird ein Beispiel angeführt:

|  |  |  |  |
| --- | --- | --- | --- |
| Bestellung |  |  |  |
| A | 1 | 3 |  |
| B | 2 | 2 |  |
| C | 3 | 1 |  |

Der Anreiz der Bestellung A die meisten neuen Blister zu geben, ist so um einiges höher.

Das Gesamtgewicht für die Bestellung kann folgendermassen angegeben werden:

|  |  |  |
| --- | --- | --- |
| Bestellung |  |  |
| A |  |  |
| B |  |  |
| C |  |  |

Durch das negative Vorzeichen beim Langzeitanteil ln soll ein möglichst kleiner Langzeitwert erzielt, also der Kurzzeitanteil maximiert werden. Wobei das Verteilen der neuen Blister bei den ältesten Bestellungen nicht so hart bestraft wird wie bei den Neuen. Wenn also z.B. nicht genügend alte Blister vorhanden sind um die Bestellung zu füllen, dann sollen vorzugsweise die ältesten Bestellungen mit neuen Blistern aufgefüllt werden und die neusten Bestellungen müssen sich mit dem Minimum an neuen Blistern begnügen.

In der Implementation wird nicht nach optimiert sondern nur nach dem Langzeitanteil (), da die Addition mit 1 für die Optimierung keine Rolle spielt, wenn sie sowieso in jedem Fall vorgenommen wird.

# Beispiel

In diesem Kapitel wird der Optimierungsvorgang an einem konkreten Beispiel aufgezeigt.

Doch zunächst einmal muss die Zielfunktion genauer untersucht werden. Diese lässt sich in Unterfunktionen aufteilen. Diese Aufteilung ist nötig, um danach die einzelnen Bestellungen separat zu optimieren, was das Beispiel um einiges einfacher gestalten wird. Die einzelnen Teile der Umformungen, die zusammengehören, wurden der Lesbarkeit halber entsprechend markiert.

Kurz gesagt kann also pro Bestellung eine Funktion definiert werden, wobei dann diese Funktionen maximiert werden.

Es liegen zwei Bestellungen zu Blistern eines Ebers mit den nachfolgenden Angaben vor.

Vorhandene Menge neuer Blister: 11

Vorhandene Menge alter Blister: 4

|  |  |  |  |
| --- | --- | --- | --- |
| Bestellung | Gewünschte Menge Bn | Mindestlangzeitanteil Ln | Funktion fn |
| A | 10 | 60% |  |
| B | 10 | 100% |  |

Es sollte klar ersichtlich sein, dass nur eine der beiden Bestellungen erfüllt werden kann. Da die Bestellung B 100% neue Blister verlangt, würden bereits 10 Stück an B gehen. Damit wären für A nicht mehr genügend Blister vorhanden. Jedoch kann entweder A oder B erfüllt werden. Die Frage stellt sich nun, welche der Bestellungen die höhere Priorität hat.

Schliesslich geht es um die Maximierung der Summe der funktionsabhängigen Variablen,

also

Anhand der Funktionsgraphen lässt sich das Beispiel besser verdeutlichen:

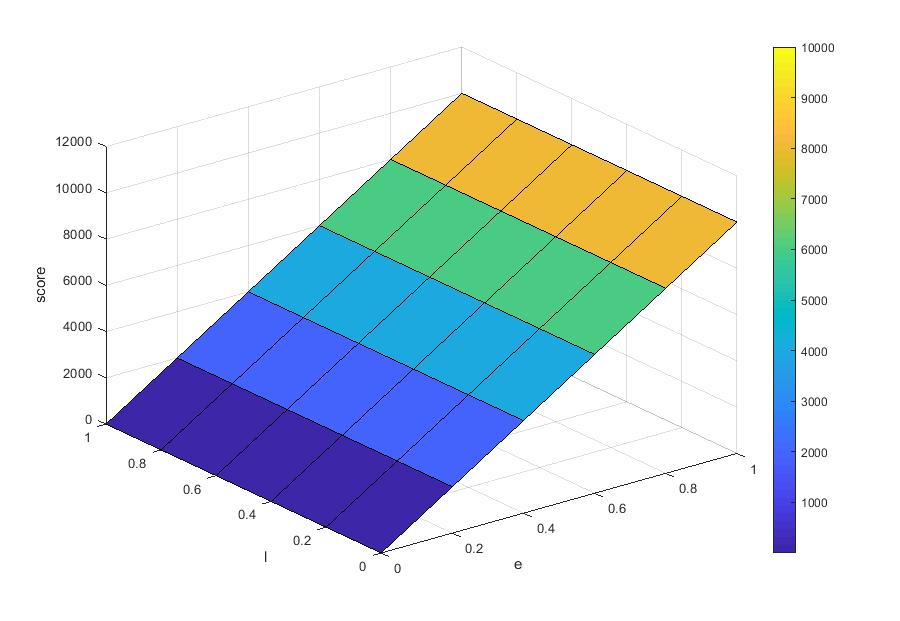


Abbildung 1 Funktionsgraph Bestellung A

Die X- sowie Y-Achse der geplotteten Funktion gehen von 0 bis 1. Dies reicht aus, da die Bestellung entweder durchgeführt (0 oder 1) wird und der Langzeitanteil beliebig zwischen 0 und 1 variieren kann. Die Z-Achse zeigt den Score der Funktion auf (Wert, der sich aus Einsetzen von e und l in die Funktion an dem jeweiligen Punkt ergibt). Dieser Score soll nun bei jeder Bestellung möglichst maximal ausfallen, was den Gesamtscore ebenfalls maximiert. Durch gewisse Einschränkungen ist nicht jeder Punkt auf der Ebene valid. Dies wird weiter unten verdeutlicht.

## Funktionsgraph unter Berücksichtigung der Nebenbedingungen

Unter Berücksichtigung der Nebenbedingungen können bereits diverse Punkte ausgeschlossen werden. Es werden nachfolgend die einzelnen Nebenbedingungen aufgeführt, welche auf eine Bestellung angewendet werden können. So gibt es weitere Nebenbedingungen, die sich nur auf die Gesamtmenge der Bestellungen anwenden lassen. Z.B. Nebenbedingung I (Alle neuen verkauften Blister dürfen die vorhandene Menge nicht übersteigen). Die Beispiele werden alle am Funktionsgraphen der Bestellung A gezeigt.

### Graph unter Berücksichtigung der Nebenbedingung enabled[0,1]

Die Variable en ist binär, kann also nur 1 oder 0 annehmen. Daher lässt sich der gültige Bereich bereits stark einschränken.

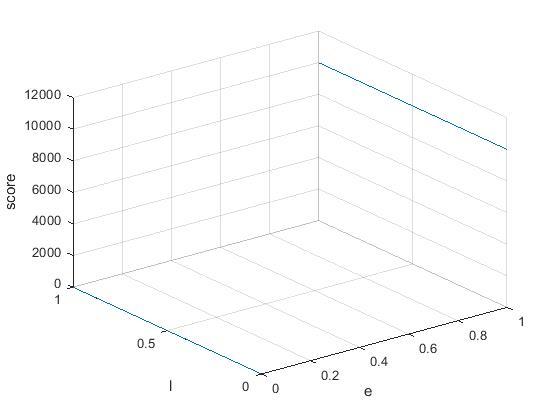


Abbildung 2 Funktionsgraph Bestellung A (enabled constraint)

Ein gültiger Wert lässt sich so nur noch auf den blau markierten Geraden finden. Ist die Bestellung ausgeschaltet, so liegt ein gültiger Wert beim Ursprung entlang der l-Achse sowie bei 1 entlang der l-Achse.

### Graph unter Berücksichtigung der Nebenbedingung VII

Wenn eine Bestellung nicht verarbeitet werden kann, so muss der Langzeitanteil 0 betragen.

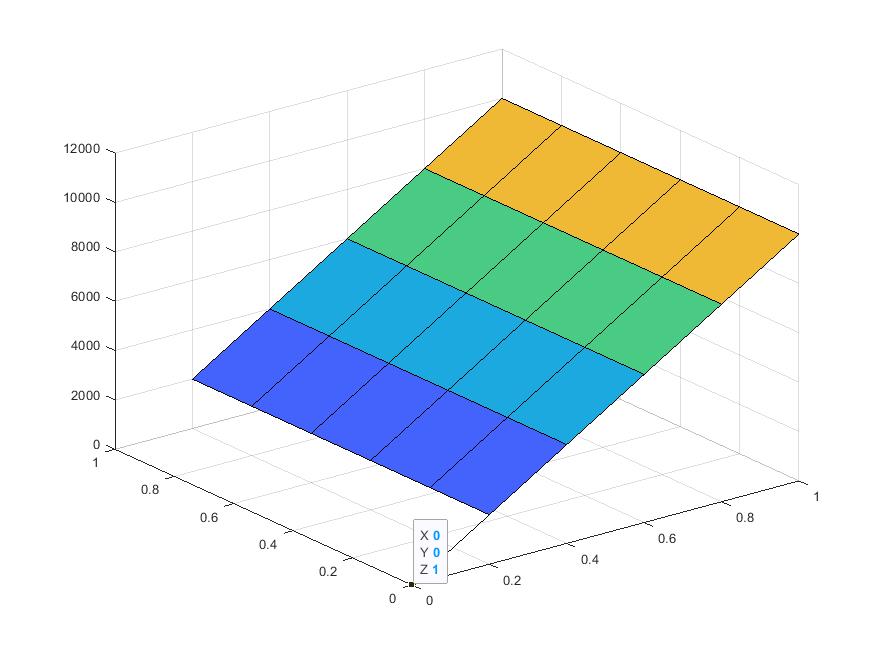


Abbildung 3 Funktionsgraph Bestellung A (Nebenbedingung VII)

Wie erkannt werden kann, ist ein Wert für die Langzeit ln nicht möglich, wenn die Bestellung ausgeschaltet ist (en = 0). Einzig der Wert 0 wird für zulässig erachtet. Man darf sich von der grossen leeren Fläche unterhalb nicht irritieren lassen. Die ist nur so gross, weil die Schritte von 0.2 gewählt wurden. Es wird effektiv nur die Gerade von (0/>0/1) bis (0/1/1) ausgeschlossen.

### Graph unter Berücksichtigung der Nebenbedingung V b

Eine Bestellung muss die Mindestvorgabe der Langzeit Ln erfüllen. So müssen bei der Bestellung A mindestens 60% der Blister aus der neuen Menge stammen. Am Graphen wird verdeutlicht, dass Langzeitwerte unter 0.6 nicht möglich sind. Hierbei wird ausser Acht gelassen, dass der Wert bei (0/0/1) weiterhin möglich sein muss (wenn Bestellung ausgeschaltet).

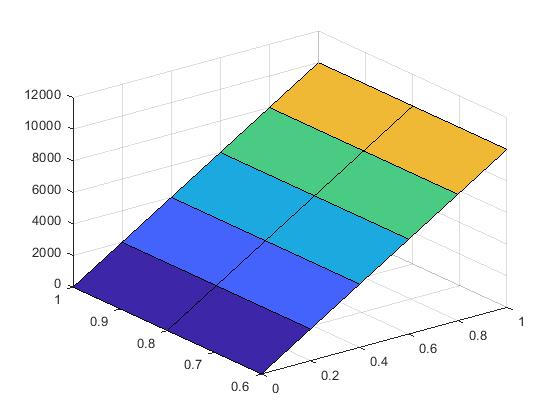


Abbildung 4 Funktionsgraph Bestellung A (Nebenbedingung V b)

### Anwendung aller Nebenbedingungen und Resultat

Werden nun alle diese Nebenbedingungen auf den Graphen angewendet, so sieht dieser die folgenden möglichen validen Werte zur Erfüllung der Bestellung vor.

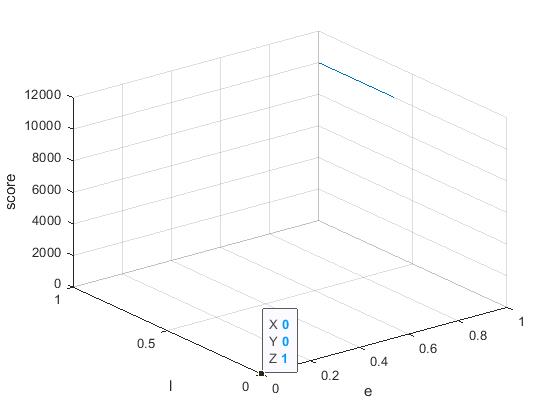


Abbildung 5 Funktionsgraph Bestellung A (alle Nebenbedingungen)

Endgültig valide Werte befinden sich nun im Punkt (0/0/1) (also ausgeschaltet) sowie auf der Geraden, die blau markiert ist. Diese Gerade beschreibt den Zustand eingeschaltet und Langzeitwerte von und über 60%.

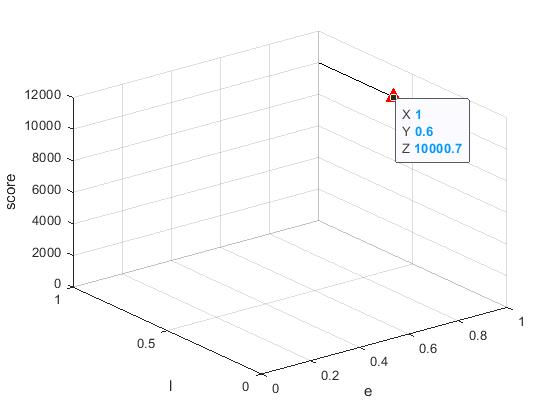


Abbildung 6 Funktionsgraph Bestellung A (Maximum )

Der Maximalwert der Bestellung A liegt nun im Modus eingeschaltet bei genau 60% Langzeitanteil und beträgt 10'000.7. Dies wurde so erwartet, da die Funktion nach Kurzzeitanteil maximiert wird, also so, dass möglichst viele alte Blister verkauft werden.

Für die Bestellung B sieht es ähnlich aus.

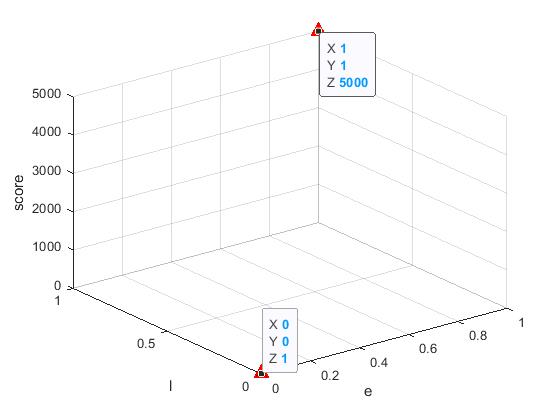


Abbildung 7 Funktionsgraph Bestellung B (Maximum)

Die Bestellung B kann lediglich einen gültigen Punkt annehmen, wenn sie eingeschaltet ist, da der Langzeitanteil bei 100% liegt. So ist das Maximum im Punkt (1/1/5000) zu finden. Der Maximalwert beträgt genau 5000. Eine weitere gültige Lösung ist die Bestellung auszuschalten (0/0/1).

Da nun die Summe der Bestellfunktionen maximiert wird, werden folgende Punkte bevorzugt:

* Bestellung A: P(1/0.6/10'000.7)
* Bestellung B: P(1/1/5000)

Dabei handelt es sich um die genannten Maxima der Bestellungen.

Nun gibt es weitere Nebenbedingungen zu beachten. Können beide Bestellungen ausgeführt werden? Sind also genügend Blister vorhanden? In diesem Beispiel kann wie bereits erwähnt nur eine Bestellung vollführt werden.

Würde nun die Bestellung B ausgeführt, liegt der Score somit bei 5000. Es wird schnell ersichtlich, dass der Wert von 10'000.7 viel lukrativer erscheint, weswegen auch die Bestellung A schliesslich bevorzugt wird. Die Bestellung B müsste somit manuell bearbeitet werden.

Nun kommen die zusätzlichen Nebenbedingungen ins Spiel. Gibt es genügend alte Blister um das Maximum der Bestellung A zu erreichen? Ja es gibt 4, was genau 60% von den gewünschten 10 sind. Der Maximalwert der Funktion f1 kann also erzielt werden.

# Anhang

**Matlab Code des Beispiels**

[variable\_e,variable\_l] = meshgrid(0:.2:1);

%Exclude invalid e points

for idx = 1:numel(variable\_e)

element = variable\_e(idx);

if (element > 0) && (element < 1)

variable\_e(idx) = nan;

end

end

%Constraint VII Exclude e = 0 wenn l != 0

for idx = 1:numel(variable\_e)

element = variable\_e(idx);

element\_l = variable\_l(idx);

if (element == 0 && element\_l > 0)

variable\_e(idx) = nan;

end

end

%Constraint V Exclude l lower than min

%Für A

min\_l = 0.6;

%Für B

%min\_l = 1;

for idx = 1:numel(variable\_l)

element = variable\_l(idx);

if (element < min\_l) && ((element > 0) || (variable\_e(idx) > 0))

variable\_l(idx) = nan;

end

end

OB\_A = 10000 .\* variable\_e + (1-variable\_l .\* (0.5));

OB\_B = 0.5\* 10000 .\* variable\_e + (1-variable\_l);

[Rmx,Cmx] = find(OB\_A == max(OB\_A(:)));

surf(variable\_e, variable\_l, OB\_A);

hold on

plot3(variable\_e(Rmx, Cmx), variable\_l(Rmx,Cmx),OB\_A(Rmx,Cmx), '^r', 'MarkerFaceColor','r', 'MarkerSize',10)

hold off

figure;

[Rmx,Cmx] = find(OB\_B == max(OB\_B(:)));

%[Lmx,Dmx] = find(OB\_B == min(OB\_B(:)));

surf(variable\_e, variable\_l, OB\_B);

hold on

plot3(variable\_e(Rmx, Cmx), variable\_l(Rmx,Cmx),OB\_B(Rmx,Cmx), '^r', 'MarkerFaceColor','r', 'MarkerSize',10)

% Plot minimum

%plot3(variable\_e(Lmx, Dmx), variable\_l(Lmx,Dmx),OB\_B(Lmx,Dmx), '^r', 'MarkerFaceColor','r', 'MarkerSize',10)

hold off