Rezept für Welfencreme, erweitert als timed Net

Jan Scholz, Lukas Kozlowski

Hochschule für angewandte Wissenschaften Hamburg,
jan.scholz2@haw-hamburg.de
lukas.kozlowski@haw-hamburg.de

Aufgabenstellung

Die uns gegebene Aufgabenstellung wurde sehr offen gewählt, es wurde verlangt, dass das erstellte Netz aus der ersten Übung erweitert wird in ein Timed Net. Auf Basis dieses neuen Netzes sollte sich dann eine Fragestellung überlegt werden die dann mittels Simulationen des Netzes beantwortet werden kann.

Unser Netz stellt ein Rezept und den dazugehörigen Herstellungsprozess dar. Da in diesem Prozess mehrere Abläufe parallel vollzogen werden können ist es sinnvoll zu erörtern wie schnell die Herstellung ist bei einer unterschiedliche Anzahl von Köchen. Betriebswirtschaftlich hat diese Frage Sinn da einerseits Zeit und andererseits Personalkosten ein wichtiger Faktor sind in der Gastronomie. Es ist zum einen schlecht wenn die Zubereitung einer Mahlzeit zu lange dauert, wenn sie frisch für einen Gast hergestellt wird. Und andererseits ist es schlecht wenn Köche aufeinander warten müssen da dann keine effektive Auslastung der Köche gewährleistet ist und weniger Geld erwirtschaftet wird im Verhältnis zu den Personalkosten.

Unter Berücksichtigung beider Aspekte formulieren wir folgende Fragestellung.

> “Wie viele Köche müssen beschäftigt werden damit mindestens 90% aller Gäste, 10min nach ihrer Bestellung, ihre Welfencreme bekommen?”

Die Arbeit in einer Küche ist natürlich wesentlich umfangreicher und wird nur zu einem kleinen Teil von unserem Petri-Netz wiedergegeben. Dieser Umstand ist bei der späteren Analyse zu berücksichtigen.

Methodik

Zunächst müssen die Transitionen, die in dem Petri-Netz aus Übung 1 erstellt wurden genauer untersucht werden. Jede Transition stellt eine Handlung dar die von einem Koch vollzogen werden kann. Zu jeder Handlung muss überlegt werden wie viel Zeit für sie benötigt wird. Renew bietet eine Verteilungsfunktion an mit der sich Zeiten zufällig bereitstellen lassen. Diese Funktion wird zunächst genauer angeschaut (siehe Abbildung 1).

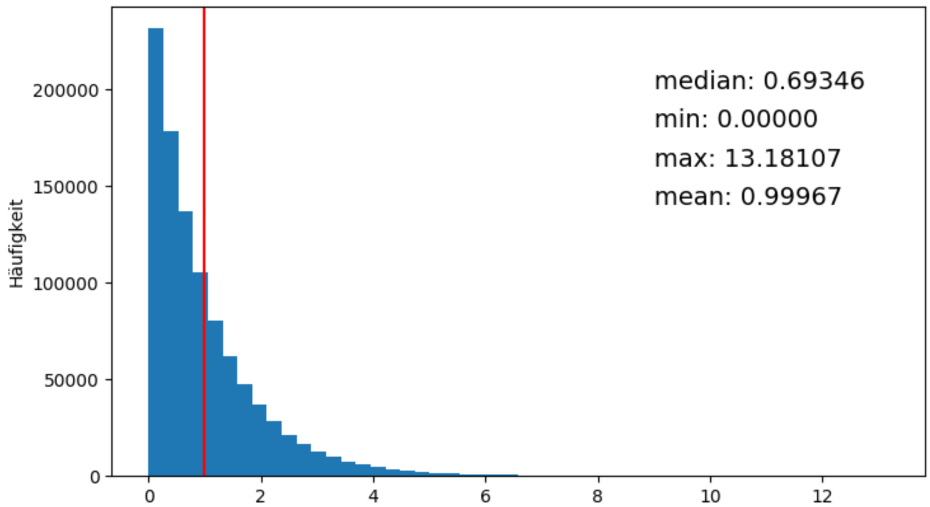

Abbildung 1: Histogramm für die Verteilungsfunktion negexp(1) von Renew nach 1Mio Wiederholungen

Wie der Name der Funktion bereits andeutet handelt es sich um eine negativ exponentiell abfallende Funktion. Aus dem gezeigten Graphen lässt sich ablesen das der Eingabewert der Funktion die durchschnittliche Zeit der Ausgabe darstellt. Die Hälfte aller Ausgabewerte ist unter 0.7 und kompakt auf einer Breite von 0.7 vertreten, die zweite Hälfte von 0.7 bis 13.8 ist wesentlich breiter verteilt mit verschwindend geringer Wahrscheinlichkeit für hohe Werte.

Ähnlich wie bei einem Wurf mehrerer Würfel ergibt sich für die Addition zweiter Funktionsaufrufe ein neuer Peak. Der neue Graph (Abbildung 2) ist nun nicht mehr konstant fallend sondern steigt zunächst bis auf sein Maximum und fällt danach ab. Ein ähnliches Verhalten wird in der Simulation ebenfalls, bei der Aneinanderreihung der Handlungen der Köche, erwartet. Diese Eigenschaft scheint intuitiv gut geeignet um die Zeit für diesen Prozess zu berechnen da bei der Herstellung eines Gerichtes einerseits Köche verschieden schnell arbeiten, was zunächst für eine Normalverteilung spricht. Allerdings können auch unvorhergesehene Dinge passieren die den Herstellungsprozess stören wie z.B. das sich ein Koch verletzt, Zutaten nicht vorbereitet sind oder ein Koch gerade eine Zigarette rauchen ist. So kann es in Einzelfällen zu wesentlich höheren Herstellungszeiten führen. Mit zunehmender Anzahl von Handlungen wird die Varianz der Zeiten im Verhältnis sinken, denn je mehr Zeiten aufaddiert werden desto geringer ist die Wahrscheinlichkeit das alle Werte sehr hoch oder sehr niedrig werden.

Es muss weiter beachtet werden, dass Handlungen einen minimalen Zeitaufwand haben. Ein Funktionsaufruf von negexp() kann auch einen Wert von 0 zurückliefern und die Handlung so direkt nach Antritt abschließen. Dieses Verhalten ist nicht gewünscht und wird verhindert dadurch, dass eine konstante

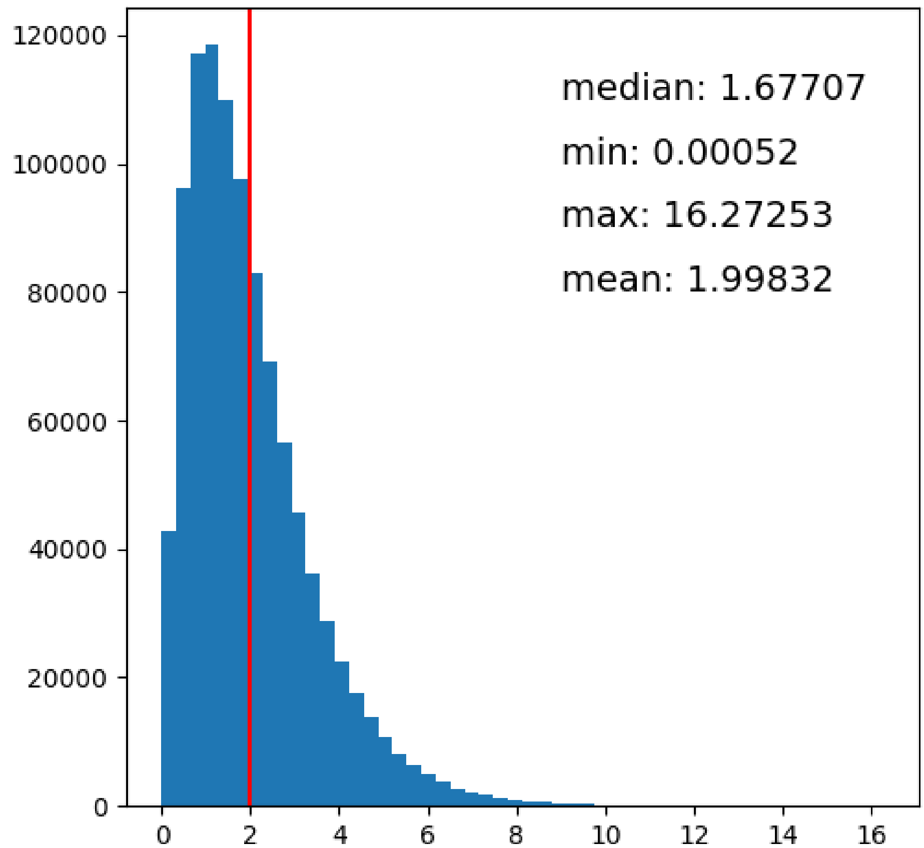

Abbildung 2: Histogramm für negexp(1) + negexp(1) für 1Mio Wiederholungen

hinzuaddiert wird, die den minimalen Zeitaufwand darstellt. Eine Zeitfunktion sie dann wie folgt aus

$$t_{tr_i} = t_{tr_{i,1}} \cdot Dist.negexp(t_{tr_{i,2}}) \tag{1}$$

$$t_{tr_i,\mu} = t_{tr_{i,1}} + t_{tr_{i,2}} \quad \text{(Durchschnitt)} \tag{2}$$

$$\widetilde{t}_{tr_i} \approx t_{tr_{i,1}} + 0.69 \cdot t_{tr_{i,2}} \quad \text{(Median)} \tag{3}$$

Für die Entwicklung des neuen timed Net's können die Köche als Ressourcen angenommen werden die jeweils nur von einer Transition zur Zeit genutzt werden können.

Wenn für alle Transitionen Zeiten hinterlegt sind kann mit den Simulationen begonnen werden. Die Simulationen werden dann für 1,2,3,4 und 6 Köche jeweils einige male wiederholt bis sich ein repräsentatives Ergebnis ablesen lässt.

Umsetzung

Tabelle 1 zeigt die von uns approximierten Zeiten für die vorhandenen Transitionen. Für alle Handlungen wird davon ausgegangen das die für die Handlung notwendigen Zutaten und Materialien dem Koch direkt vorliegen und der Koch sofort mit dem Vorgang beginnen kann.

Tabelle 1: Zeiten für Handlungen/Transitionen

Handlung / Transition	Zeit in s
Vanilleschote aufschneiden	3 + Dist.negexp(1)
Mark aufkochen	180 + Dist.negexp(5)
Milch & Stärke verrühren	5 + Dist.negexp(4)
Zitrone & Wein mischen	5 + Dist.negexp(1)
Weinsoße schaumig rühren	40 + Dist.negexp(5)
Milch/Stärke & Vanillemilch verrühren	5 + Dist.negexp(1)
Vanillecreme aufkochen	60 + Dist.negexp(10)
Vanillecreme abkühlen lassen	260 + Dist.negexp(40)
Ei trennen	5 + Dist.negexp(1)
Eiweiss schlagen	20 + Dist.negexp(4)
Schnee unter Vanillecreme unterheben	5 + Dist.negexp(5)
Vanillecreme + Schnee abfüllen	2 + Dist.negexp(2)
Vanillecreme mit heißer Weinsoße auffüllen	2 + Dist.negexp(2)

Analyse

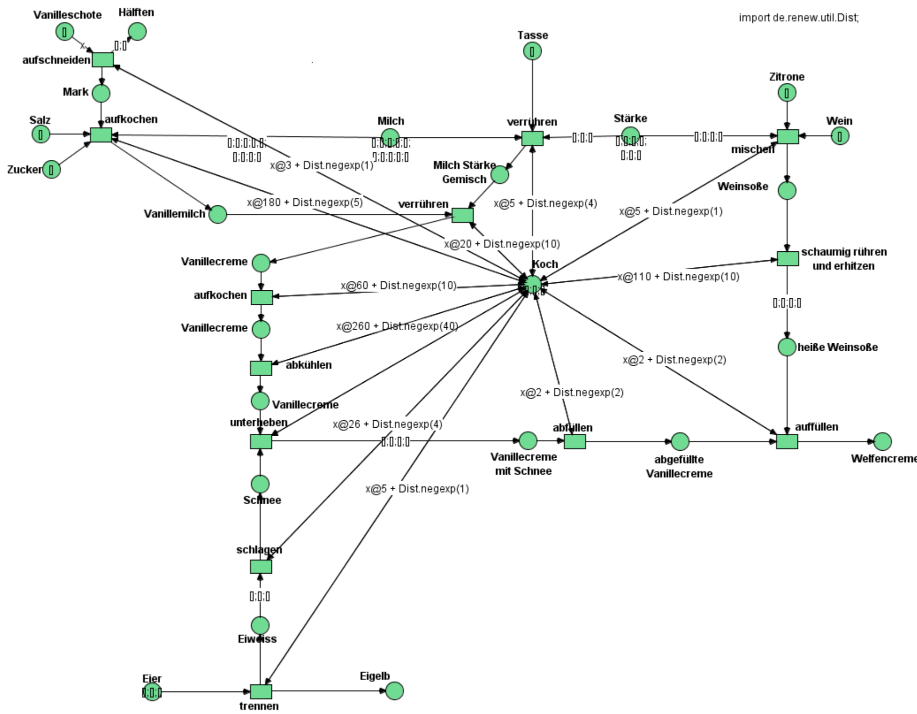

Abbildung 3: Das auf Basis der Rezeptur erstellte Petrinetz, erweitert zum timed Net

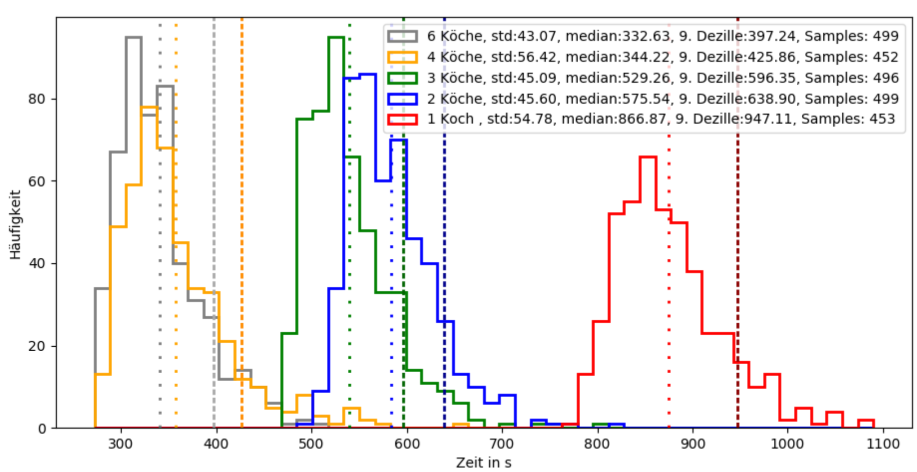

Abbildung 4: Histogramm: Die Simulation des Petrinetzes mit verschiedener Anzahl Köchen, gepunktet der korrespondierende Median, dunkel gestrichelt 9. Dezille