Untersuchung von Strategien zur Koordination von Flugzeugen an eine oder zwei Landebahnen

Mario Rohrhofer und Sebastian Achtert

2. August 2010

1 Einleitung

Im modernen Flugverkehr gewinnt die effiziente Nutzung der Lande- bzw. Startbahnen, auf internationalen Flughäfen immer mehr an Bedeutung. Während die Flugzeuge auf eine freie Landebahn warten, werden sie oft in eine „Warteschlange“ geschickt und müssen zusätzlich einen Sicherheitsabstand zum voraus fliegenden Flugzeug einhalten, da dieses Luftturbulenzen auslöst. Diese Abstände (Seperation) hängen vom jeweiligen Typ des Flugzeuges und von dessen Gewicht ab. Augenscheinlich ist nun, dass hier versucht wird die Wartezeit für Passagiere zu minimieren, was Hand in Hand mit dem Versuch möglichst Treibstoff zu sparen geht.

Dieses Problem kann als spezielles Warteschlangensystem mit anfliegenden Flugzeugen als Kunden verschiedener Typen beschrieben werden. Die Bedienzeit des n-ten Flugzeuges ist nun also die Seperationszeit zwischen dem n-ten und dem $n+1$-ten Flugzeuges. Die Bedienung beginnt, wenn das n-te Flugzeug aufsetzt („touchdown“) und endet frühestens wenn das $n+1$-te landen könnte.

Die Ankunftszeit ist ein bestimmter Punkt, an dem das Flugzeug in den Bereich des Towers und somit der Flugsicherung („traffic control“) kommt. Im Modell mit zwei Landebahnen wird das Flugzeug dann auf eine der beiden gelotst. Wir werden in diesem Modell die Zeit von dem Eintritt in den Raum der Flugsicherung bis zum Landeanflug, also der Bedienung, betrachten. Wenn also das Flugzeug den Eintrittspunkt überschreitet und eine Landebahn wäre frei, so könnte es ohne weiteren Zeitverlust landen. Also ist die Wartezeit, die Zeit vom Eintrittspunkt bis zur Landung. Zusätzlich werden wir uns auch noch die Stabilität unseres Modells und die möglichen Grenzen für die durchschnittliche Wartezeit ansehen.

Wir werden den allgemeinen Ansatz wählen, dass die Ankunftszeiten durch einen Poisson-Prozess beschrieben werden. Im Falle zweier Landebahnen versuchen wir abermals die durchschnittlichen Wartezeiten zu minimieren, wodurch wir an guten Strategien zur Verteilung der ankommenden Flugzeuge auf die Landebahnen interessiert sind. Unter diesen Strategien befinden sich „Coin Flipping“, „Round Robin“ oder „Join-the-least-load“.

Als Komplexifizierung des Modells könnten noch Treibstoffmangel, oder Starts zwischen den einzelnen Landungen mit einbezogen werden. Dies werden wir in unserer Konklusion kurz ansprechen.

2 Grundbegriffe der Warteschlangentheorie

Den Begriff „Warteschlange“ kennt man aus dem täglichen Leben. Personen, die zum Beispiel vor den Kassen in einem Supermarkt oder an den Schaltern von Bahnhöfen warten, bilden eine Warteschlange. Aber auch Autos vor einer Waschanlage können ein Warteschlangensystem bilden.

Die Zeit zwischen dem Eintreffen von zwei Forderungen und die Länge der Bedienzeit einer Forderung sind zufällige Größen. Die Aufgabe der Statistik besteht nun darin, ihre Verteilungen zu bestimmen. Mit Hilfe dieser Verteilung lassen sich dann weitere, für das Warteschlangensystem charakteristische Größen ermitteln. So etwa die Anzahl der Forderungen im System bzw. die Zeit, die eine Forderung bis zu ihrer Bedienung warten muss. Dabei zeigt sich, dass eine große Klasse von Warteschlangensystemen mit Hilfe der Theorie der Poisson-Prozesse beschrieben werden kann. Geburts- und Todesprozesse sind Beispiele für solche Prozesse. Dies ist jedoch nicht für alle aus der Praxis abstrahierten Modelle für Warteschlangen der Fall.

2.1 Der Poisson-Prozess

Wir werden nun den Begriff des Poisson-Prozesses näher erläutern. Unter *Wikipedia* findet man dazu folgende Definition:

„Der Poisson-Prozess ist ein Erneuerungsprozess, dessen Zuwächse unabhängig und poissonverteilt mit Parameter λ sind. λ wird auch als Intensität des Prozesses bezeichnet, da pro Zeiteinheit genau λ Sprünge erwartet werden (Erwartungswert der Poissonverteilung ist ebenfalls λ). Die Höhe jedes Sprunges ist eins, die Zeiten zwischen den Sprüngen sind exponentialverteilt. Der Poisson-Prozess ist also ein diskreter Prozess in stetiger (d.h. kontinuierlicher) Zeit.“

Definition 2.1 *Ein stochastischer Prozess über dem Wahrscheinlichkeitsraum* $[\Omega; \mathfrak{A}; \mathbb{P}]$ *heißt (homogener) Poisson-Prozess* $P_{\lambda,t}$ *mit Intensität* λ *und* $t \in [0; \infty)$, *falls folgende drei Bedingungen erfüllt sind:*

- $P_{\lambda,0} = 0 \ \ (\mathbb{P} - f.s.)$
- $P_{\lambda,t} - P_{\lambda,s} \sim \mathcal{P}_{\lambda\cdot(t-s)} \quad \forall\, s < t$. *Dabei bezeichnet* $\mathcal{P}_{\lambda\cdot(t-s)}$ *die Poisson-Verteilung mit Parameter* $\lambda \cdot (t-s)$.
- *Sei für* $n \in \mathbb{N}$ *eine Folge* $0 < t_1 < \dots < t_n$ *gegeben. Dann ist die Familie* $\langle P_{\lambda,t_i} - P_{\lambda,t_{i-1}} \mid 2 \leq i \leq n \rangle$ *von Zufallsvariablen stochastisch unabhängig.*

In der folgenden Formel formulieren wir eine Eigenschaft der Poisson-Verteilung. Sie besagt, dass die Wahrscheinlichkeit, mit der die Bearbeitung einer Forderung zum Zeitpunkt t in der Zukunft abgeschlossen sein wird, nicht davon abhängt, wie lange sie schon bearbeitet wurde, sondern nur von der Differenz $t - t_0$, denn für $t > t_0 > 0$ gilt

$$\begin{aligned} P(S_n < t \mid S_n \geq t_0) &= \frac{P(t_0 \leq S_n < t)}{P(S_n \geq t_0)} \\ &= \frac{B(t) - B(t_0)}{1 - B(t_0)} \\ &= 1 - e^{-\mu(t-t_0)}. \end{aligned} \tag{2.1}$$

D.h., an Stelle der Gesamtzeit t ist nur die Zeitdifferenz $t - t_0$ interessant. Der Prozess „vergisst“ also die Entwicklung bis zum Zeitpunkt t_0, beginnt dann erneut und „schaut“ in die Zukunft auf die verbleibende Restzeit $t - t_0$. Die negative Exponentialverteilung ist die einzige stetige Verteilung mit dieser Eigenschaft.

Das Superpositionsprinzip kann man auch hier anwenden. Ist N_t ein Poisson-Prozess mit Intensität λ sowie $Y_1, Y_2, \ldots$ unabhängige, identisch verteilte Zufallsvariablen unabhängig von N_t, so wird der stochastische Prozess $X_t := \sum_{n=1}^{N_t} Y_n$ als „zusammengesetzter Poisson-Prozess“ bezeichnet. Wie der ursprüngliche Poisson-Prozess ist auch X ein solcher.

2.2 Definition eines Warteschlangensystems

Der Ankunftsprozess beschreibt die Art und Weise, in der die Forderungen in dem Warteschlangensystem eintreffen. T_n ist die Zeit, die zwischen der Ankunft der $(n-1)$-ten und der n-ten Forderung vergeht (Zwischenankunftszeit). Diese Zwischenankunftszeiten $T_1,\ T_2,\ T_3, \ldots$ können für unser Modell als unabhängig und identisch verteilt angesehen werden.

Die Zeit, die bis zur Ankunft der n-ten Forderung vergeht, ist $S_n := \sum_{k=1}^{n} T_k$ bzw. $S_0 := 0$. $N(t)$ sei die Anzahl der Ankünfte bis zum Zeitpunkt $t \geq 0$. Dieser Zählprozess heißt Ankunftsprozess, der im Zeitpunkt $t = 0$ mit $N(0) = 0$ startet. Weiter gilt

$$P(N(t) = n) = P(S_n \leq t \ < \ S_{n+1}). \tag{2.2}$$

D.h., die Wahrscheinlichkeit für das Ereignis „n Ankünfte bis zum Zeitpunkt t“ ist gleich der Wahrscheinlichkeit des Ergebnisses „n-te Ankunftszeit kleiner als oder gleich t und $(n+1)$-te Ankunftszeit größer als t“. Bei einer großen Anzahl von Warteschlangenmodellen ist die Annahme eines poissonverteilten Forderungseingangs statistisch gesichert. $N(t)$ besitzt also die Wahrscheinlichkeitsverteilung

$$P(N(t) = n) = \frac{(\lambda t)^n}{n!} \cdot e^{-\lambda t} \text{ für } t \geq 0. \tag{2.3}$$

Die Verteilungsfunktion $T(t)$ der T_n erhält man nun aus

$$\begin{aligned} P(T_n \geq t) &= P(\{\text{keine Ankunft im Intervall } [S_{n-1}, S_{n-1} + t)\}) \\ &= P(N(S_{n-1} + t - S_{n-1}) = 0) \\ &= P(N(t) = 0) = e^{-\lambda t} \end{aligned} \tag{2.4}$$

und ist gegeben durch

$$T(t) = \begin{cases} 0 & \text{für } t \leq 0 \\ 1 - e^{-\lambda t} & \text{für } t > 0. \end{cases} \tag{2.5}$$

Die T_n sind also negativ exponentialverteilt mit Mittelwert $\frac{1}{\lambda}$. Wir werden diese Erkenntnis weiter unten benutzen.

Die sogenannte Warteschlangendisziplin beschreibt die Reihenfolge, in der die Forderungen am Schalter vom Bedienmechanismus abgearbeitet werden. Die wichtigesten Warteschlangendisziplinen sind:

1. FIFO: „first in, first out": Forderungen werden in der Reihenfolge ihres Ankommens bedient.

2. LIFO: „last in, first out": Die Forderung, die zuletzt angekommen ist, wird zuerst bedient.

3. SIRO: „service in random order": Forderungen werden in einer Reihenfolge abgearbeitet, die einem Zufallsmechanismus unterliegt.

4. PRI: „priority": Forderungen werden Prioritäten zugeteilt und entsprechend ihrer Reihenfolge abgearbeitet. Die Bearbeitung innerhalb einer Priorität wird nach FIFO durchgeführt.

In dem Modell mit einer Landebahn wird ausschließlich die FIFO-Disziplin angewendet. Bei einer möglichen Komplexifizierung mit Beachtung der Treibstoffmenge würde dann eine PRI-Disziplin vorliegen.

Beim Warteraum besteht die Möglichkeit, dass er physikalisch begrenzt oder gar nicht vorhanden ist und daher nur eine endliche Anzahl von Forderungen vor dem Schalter warten kann. Wenn der Warteraum besetzt ist, müssen neue hinzukommende Forderungen abgewiesen werden und gehen somit dem Warteschlangensystem verloren. Die zweite Möglichkeit ist ein unendlich großer Warteraum, also ohne physikalische Begrenzung.

Der Bedienungsmechanismus besteht aus $1 \leq s \leq \infty$ Schaltern, an denen die Forderungen bedient werden. Angenommen die Bedienzeit B_n seien von einander unabhängige Zufallsvariablen mit der selben Verteilung und unabhängig von den Zwischenankunftszeiten. Ihre gemeinesame Verteilungsfunktion $B(t)$ lautet dann

$$B(t) = \begin{cases} 0 & \text{für } t \leq 0 \\ P(B_n < t) & \text{für } t > 0. \end{cases} \tag{2.6}$$

Vielen Warteschlangenmodellen werden negativ exponentialverteilte Bedienzeiten zugeordnet, also ergibt sich

$$B(t) = \begin{cases} 0 & \text{für } t \leq 0 \\ 1 - e^{-\mu t} & \text{für } t > 0. \end{cases} \tag{2.7}$$

Der stochastische Prozess $\{L(t), t \geq 0\}$, der die Anzahl der sich zum Zeitpunkt t im Warteschlangensystem befindlichen Forderungen angibt, ist ebenfalls ein Poisson-Prozess. Die für die Praxis interessanten zeitunabhängigen (stationären) Wahrscheinlichkeitsverteilungen lauten

$$l_n := \lim_{t \to \infty} l_n(t) := \lim_{t \to \infty} P(L(t) = n) =: P(L = n). \tag{2.8}$$

Dabei gibt die Zufallsvariable L mit der Wahrscheinlichkeitsverteilung l_n die Anzahl der Forderungen in einem stationären Warteschlangensystem an. Als nächstes definieren wir die Verkehrsintensität durch

$$\rho := \frac{\text{mittlere Bedienzeit}}{\text{mittlere Zwischenankunftszeit}}.$$

Zum Abschluss dieses Kapitels soll noch die Bezeichnung der Warteschlangensysteme festgelegt werden. Die Warteschlange wird durch folgende vier Symbole beschrieben:

$$A \mid B \mid X \mid Y.$$

Diese haben folgende Bedeutungen:

A ... Verteilung der Zwischenankunftszeiten

Symbol	M	(Exponentialverteilung, M steht für die Markov-Eigenschaft der negativen Exponentialverteilung)
	D	(Deterministisch)
	G	(allgemeine Verteilung)

B ... Verteilung der Bedienzeiten

Symbol	M	(siehe oben)
	D	(siehe oben)
	G	(siehe oben)

X ... Anzahl der parallelen Schalter

Symbol $1, 2, ..., \infty$

Y ... Beschränkung der Systemkapazität

Symbol $1, 2, ..., \infty$

3 Modell mit einer Landebahn

Ankunftsprozess

Wir nehmen an, dass die Ankunftszeiten $(S_n)_{n\geq 1}$ der Flugzeuge die Form eines Geburtsprozesses mit unabhängigen Zwischenankunftszeiten $T_n := S_n - S_{n-1}$ für $n \geq 1$ und $S_0 = T_0 = 0$ besitzen. Zusätzlich sei die Intensität des Ankunftsprozesses beschrieben durch $\lambda = (\mathbb{E}T_1)^{-1} > 0$ und $(N_t)_{t\geq 0}$ sei die Anzahl der Ankünfte bis zum Zeitpunkt t. Wir nehmen an, dass die erste Ankunft zur Zeit $S_1 = T_1$ stattfindet.

Für unser Modell betrachten wir die Typen der ankommenden Flugzeuge als zufällig, da diese trotz geregeltem Flugplan durch Störungen (z.B. Wetter) nicht determiniert sind. Die Wahrscheinlichkeit, dass ein ankommendes Flugzeug vom Typ $j \in J$ ist, sei $p_j \in (0,1)$. Weiter seien die Typen unabhängig voneinander und von der Ankunftszeit. D.h., wenn J_n den Typ des n-ten Flugzeuges beschreibt, dann nehmen wir an, dass gilt

$$P(J_n = i) = p_i \tag{3.1}$$

und nennen $p := (p_1, ..., p_{|J|})$ den „Traffic Mix“.

Da der Ankunftsprozess ein Poisson-Prozess ist, können wir ihn als Superposition von unabhängigen Poisson-Prozessen $(N_{j,t})_{t\geq 0},\ j \in J$, mit der jeweiligen Intensität $\lambda_j := p_j\lambda$ darstellen.

Separationszeit

Ein Flugzeug vom Typ $i \in J$ verursacht Luftverwirblungen, so dass das folgende Flugzeug vom Typ $j \in J$ einen Minimalabstand einhalten muss, welchen wir als Separationszeit $b(i,j)$ bezeichnen. Somit kann die Landung des folgenden Flugzeuges frühestens $b(i,j)$ Zeiteinheiten nach Landung des vorausfliegenden Flugzeuges stattfinden. Im Folgenden fassen wir die Separationszeiten zur Matrix $(b(i,j))_{i,j\in J}$ zusammen.

Bedienzeit

Betrachten wir die Flugzeuge als wartende Kunden an einer Kasse, so ergibt sich die Bedienzeit des n-ten Kunden als Separationszeit zum nächsten. Dies ist die n-te Bedienzeit

$$B_n := b(J_n, J_{n+1}). \tag{3.2}$$

Wenn wir annehmen, dass das n-te Flugzeug vom Typ $J_n = i$, das vorige vom Typ $J_{n-1} = k$ und der Bedienmechanismus zur Aunkunftszeit S_n des n-ten Flugzeuges leer ist, die Seperationszeit $b(k,i)$ also bereits verstrichen ist, dann findet die Landung ohne Wartezeit statt. Die Bedienzeit B_n ist zum Zeitpunkt der Landung noch unbekannt, da sie auch vom darauf folgenden Flugzeug des Typs $J_{n+1} = j$ abhängt und somit erst zum Ankunftszeitpunkt S_{n+1} ermittelt werden kann. Sollte die Seperationszeit $b(i,j)$ noch nicht verstrichen sein, so ergibt sich die Wartezeit W_{n+1}.

Die Bedienzeiten sind identisch verteilt und bilden einen stationären Prozess, werden jedoch im Allgemeinen nicht unabhängig sein. Jedoch sind die Zwischenankunftszeiten und die Bedienzeiten von einander unabhängig.

Weiter ergibt sich die Verkehrsintensität wie in Kaptiel 2.1 definiert als

$$\rho = \frac{\mathbb{E}B_1}{\mathbb{E}T_1} \tag{3.3}$$

Wartezeit

Die Wartezeit eines Flugzeuges wird als die Zeit zwischen der Ankunftszeit am Flughafen und der Landung („touchdown“) gemessen. Wie in der Einleitung angekündigt, werden wir die Flugzeit von der Ankunft bis zur Landebahn nicht berücksichtigen. Wenn also das Flugzeug den Eintrittspunkt überschreitet und eine Landebahn frei wäre, so könnte es ohne Zeitverlust landen. Wir definieren die Wartezeit rekursiv durch

$$\begin{aligned} W_1 &:= 0 \\ W_{n+1} &:= \max\{0, W_n + B_n - T_{n+1}\}. \end{aligned} \tag{3.4}$$

Jedoch wird das Hauptaugenmerk auf der durchschnittliche Wartezeit liegen, also

$$\limsup_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} W_n. \tag{3.5}$$

Stabilität

Die Folge der Wartezeiten $(W_n)_{n\geq 1}$ heißt stabil, wenn es eine nicht negative, reelle Zufallsvariable W_∞ gibt, sodass

$$\lim_{n\to\infty} W_n = W_\infty \tag{3.6}$$

in der Verteilung gilt. Mit gegebenen Seperationszeiten hängt schließlich die Stabilität der Wartezeiten von der Ankunftsintensität λ ab. Somit sind wir vor allem am Supremum $\bar{\lambda}$ aller λ interessiert, um stabile Wartezeiten zu garantieren. Wir nennen $\bar{\lambda}$ die *absolute Kapazität.*

Wir werden nun einige wenige Resultate über stationäre Wartezeiten in Warteschlangensystemen mit unabhängigen Bedienzeiten geben. Es sei D eine $|J| \times |J|$-Matrix mit den Einträgen $d(i,j) := p_i\, p_j\, b(i,j)$. Dann gilt

$$\mathbb{E}B_1 \;=\; \mathbb{E}b(J_1, J_2) \;=\; \sum_{i,j\in J} p_i\, p_j\, b(i,j) = \mathbf{1}^T D \mathbf{1}, \tag{3.7}$$

wobei hier $\mathbf{1}$ gleich dem Vektor mit Dimension $|J|$ mit nur der 1 als Einträgen entspricht. Aus

$$\mathbb{E}T_1 \;=\; \frac{1}{\lambda} \tag{3.8}$$

erhalten wir

$$\rho \;=\; \lambda \mathbf{1}^T D \mathbf{1}. \tag{3.9}$$

Es gilt:

$$(W_n)_{n\geq 1} \text{ ist stabil genau dann, wenn } \rho < 1. \tag{3.10}$$

Dies geht offensichtlich aus der Definition von ρ hervor, denn wenn die Bedienzeiten kürzer sind als die Zwischenankunftszeiten, wird es nie zu einer Überfüllung des Warteraumes kommen, da sich das Sytem immer wieder leert. Die Wartezeiten gehen somit nicht gegen Unendlich.

Somit haben wir die notwendigen Größen für das Modell mit einer Landebahn hergeleitet. Im Folgenden werden wir dieses Modell auf zwei Landebahnen erweitern.

4 Modell mit zwei Landebahnen

Nachdem wir nun das Modell mit einer Landebahn untersucht haben, wollen wir uns dem komplexeren Fall mit zwei Landebahnen, die wir I und II nennen wollen, hingeben, in dem ankommende Flugzeuge mit Hilfe einer Strategie δ auf eine der beiden gelotst werden.

Nun werden wir wieder $T_1, T_2, ...$ als Zwischenankunftszeiten an dem Punkt annehmen, an dem entschieden wird, welche Landebahn anzufliegen ist. $p := (p_1, ..., p_{|J|})$ nennen wir erneut den „Traffic Mix“ des ankommenden Stroms. Dieser wird nun in zwei seperate Ströme mittels der Strategie δ aufgeteilt. Seien nun also die Zwischenankunftszeiten des Stroms der Landebahn $a \in \{I, II\}$ gegeben durch $T_1^a, T_2^a, ...$. Wir werden alle benötigten Größen durch diesen oberen Index der Landebahn $a \in \{I, II\}$ zuordnen. Somit folgt zum Beispiel für die Landebahn I die Bedienzeiten B_m^I, die Wartezeiten W_m^I und W_∞^I, $\lambda^I := 1/\mathbb{E}T_1^I$ und die Verkehrsintensität ρ^I, falls die Zwischenankunftszeiten $T_1^I, T_2^I, ...$ unabhängig voneinander sind. Kurz bemerken wollen wir noch, dass der Index m jeweils separat die Ankünfte der beiden Landebahnen I und II zählt.

Wir nehmen nun wieder an, dass die Zwischenankunftszeiten $T_1, T_2, ...$ unabhängig voneinander und exponentialverteilt sind, also der Ankunftsprozess ein Poisson-Prozess $(N(t))_{t\geq 0}$ mit der Ankunftsrate λ ist.

4.1 Strategien mit zufälliger Aufteilung

Zuerst werden wir Strategien betrachten, die nur vom Typ des ankommenden Flugzeuges abhängen, und werden den „Traffic Mix“ zufällig auf die zwei Landebahnen aufteilen.

Es sei Δ die Menge aller zufälligen Entscheidungsregeln $\delta = (\delta_j)_{j\in J}$ mit $\delta_j \in [0,1]$ als Wahrscheinlichkeit, dass ein Flugzeug vom Typ j zur Landebahn I gewiesen wird und $1 - \delta_j$, dass es zur anderen geleitet wird. Mit A_n bezeichnen wir die Entscheidung für das n-te Flugzeug. Es folgt

$$P(A_n = I \mid J_n = j) \;=\; \delta_j \text{ für } j \in J \text{ und } n \geq 1. \tag{4.1}$$

und die A_n sind unabhängig.

Der große Vorteil liegt nun darin, dass uns zwei unanhängige Warteschlangen an den Landebahnen zur Verfügung stehen. Die Entscheidungsregel $\delta \in \Delta$ teilt also den Ankunftsprozess $(N(t))_{t\geq 0}$ in zwei unabhängige Poisson-Prozesse $(N(t)^I)_{t\geq 0}$ und $(N(t)^{II})_{t\geq 0}$ mit deren Intensitäten

$$\lambda^I \ := \ \lambda \sum_{j \in J} p_j \delta_j \ = \ \lambda p^T \delta \tag{4.2}$$

und

$$\lambda^{II} \ := \ \lambda(1 - p^T \delta). \tag{4.3}$$

Wir nehmen $p^T\delta \in (0,1)$ und $\lambda^I > 0, \lambda^{II} > 0$ an. Somit sind die zwei Landebahnen voneinander unabhängig und verhalten sich - jede für sich - wie im Modell mit einer Landebahn. Die Wahrscheinlichkeit, dass ein Flugzeug an der Landebahn I bzw. II vom Typ $i \in J$ ist nun gegeben durch

$$p_i^I := \frac{p_i \delta_i}{p^T \delta} \quad \text{bzw.} \quad p_i^{II} := \frac{p_i(1-\delta_i)}{p^T(\mathbf{1} - \delta)}. \tag{4.4}$$

p^I und p^{II} seien nun die zwei lokalen „Traffic Mixes“, also für die beiden Landebahnen getrennt. Sei B_m^a die Bedienzeit des m-ten Flugzeuges an der Landebahn $a \in \{I, II\}$, dann erhalten wir

$$\mathbb{E}B_1^I \ = \ \sum_{i,j \in J} p_i^I p_j^I b(i,j) = \frac{\delta^T D \delta}{(p^T\delta)^2} \quad \text{und} \quad \mathbb{E}B_1^{II} \ = \ \frac{(\mathbf{1} - \delta^T) D (\mathbf{1} - \delta)}{(1 - p^T\delta)^2}. \tag{4.5}$$

Die lokalen Verkehrsintensitäten ρ^I und ρ^{II} ergeben sich somit als

$$\rho^I \ = \ \lambda \frac{\delta^T D \delta}{(p^T \delta)} \quad \text{und} \quad \rho^{II} \ = \ \lambda \frac{(\mathbf{1} - \delta^T) D (\mathbf{1} - \delta)}{(1 - p^T \delta)}. \tag{4.6}$$

Das folgende Korollar wird uns nun die Konditionen der simultanen Stabilitäten auf beiden Landebahnen geben.

Korollar 4.1 *Unter der Strategie $(\delta_j)_{j \in J} \in \Delta$ führt uns die Ankunftsrate λ zu simultanen Stabilitäten genau dann, wenn*

$$\begin{aligned} \lambda < \bar{\lambda}(\delta) \ &:= \ \min \left\{ \frac{p^T\delta}{\delta^T D \delta}, \frac{(1 - p^T\delta)}{(\mathbf{1} - \delta^T) D (\mathbf{1} - \delta)} \right\} \\ &= \ \left(\max\{ p^T \delta \ \mathbb{E}B_1^I, (1 - p^T\delta) \ \mathbb{E}B_1^{II} \} \right)^{-1}. \end{aligned} \tag{4.7}$$

Beweis. Die Stabilität beider Landebahnen ist äuquivalent zu $\max\{\rho^I, \rho^{II}\} < 1$. Somit erhalten wir aus (4.5): $\frac{p^T\delta}{\delta^T D \delta} = (p^T \delta \ \mathbb{E}B_1^I)^{-1}$ und analog für $a = II$. □

Wir nennen $\bar{\lambda}(\delta)$, definiert in (4.7) die absolute Kapazität der Landebahnen bei Anwendung der Strategie δ.

4.2 Fair Coin Flipping

Diese Strategie kann man sich folgendermaßen vorstellen: Man nimmt eine faire Münze für alle möglichen Typen von Flugzeugen, um zu entscheiden, auf welche Landebahn sie gelotst werden. Daraus erhalten wir die einfache Strategie $\delta^{Coin} \in \Delta$ mit

$$\delta_i^{Coin} = \frac{1}{2} \text{ für } i \in J. \tag{4.8}$$

In diesem Fall erhalten wir

$$p_i^I = \frac{1/2\ p_i}{\sum_{j\in J} 1/2\ p_j} = p_i = p_i^{II}.$$

Also gilt $p^T \delta^{Coin} = 1/2 = 1 - p^T \delta^{Coin}$ und $\lambda^I = \lambda^{II} = \lambda/2$. Weiters können wir

$$\mathbb{E}B_1^I \;=\; \mathbb{E}B_1^{II} \;=\; \mathbf{1}^T D\mathbf{1} \quad \text{und} \quad \rho^I \;=\; \rho^I \;=\; \lambda\mathbf{1}^T D\mathbf{1}/2 \tag{4.9}$$

bestimmen, was uns zur absoluten Kapazität von

$$\bar{\lambda}(\delta^{Coin}) \;=\; \frac{2}{\mathbf{1}^T D\mathbf{1}} \tag{4.10}$$

führt. Mit dieser Strategie erhalten wir also die doppelte Kapazität des Modells mit nur einer Landebahn. Dies ist auf jeden Fall das mindeste, was wir von einer Strategie bei zwei Landebahnen erwarten und wir werden auch in unseren Simulationsergebnissen im Anschluss sehen, dass δ^{Coin} eine wirklich schlechte Strategie ist.

4.3 Die Round Robin Strategie

Eine weitere wirklich einfache Strategie ist die mit δ^{RR} bezeichnete Round Robin Strategie, die nach dem Schema $1, 3, 5, ...$ nach Landebahn I und $2, 4, 6, ...$ nach Landebahn II die Flugzeuge verteilt.

Auf den beiden lokalen Landebahnen ergibt sich

$$\begin{aligned} T_m^I &= T_{2m-2} + T_{2m-1}, \quad & T_m^{II} &= T_{2m-1} + T_{2m} \quad \text{und} \\ J_m^I &= J_{2m-1}, \quad & J_m^I &= J_{2m}. \end{aligned} \tag{4.11}$$

Somit sind die Ankunftsprozesse auf den beiden Landebahnen „Geburtsprozesse“, die jedoch nicht mehr voneinander unabhängig sind. Auf Grund der unabhängigen Wahrscheinlichkeiten der Typen $(J_n)_{n\geq 1}$, haben die Teilfolgen $(J_n^I)_{n\geq 1}$ und $(J_n^{II})_{n\geq 1}$ die selbe Verteilung und daher folgt für δ^{RR}

$$p_i^I = p_i = p_i^{II} \text{ für } i \in J \quad \text{und} \quad \mathbb{E}B_m^I \; = \; \mathbb{E}B_m^{II} \; = \; \mathbf{1}^T D \mathbf{1}. \tag{4.12}$$

Die Zwischenankunftszeiten $T_1^a, T_2^a, ...$ sind unabhängig und identisch verteilt und unabhängig von den stationären Folgen $B_1^a, B_2^a, ...$ Wir erhalten nun die Stabilität der Landebahn a genau dann, wenn

$$1 \; > \; \rho^a \; = \; \frac{\mathbb{E}B_1^a}{\mathbb{E}T_1^a} = \frac{\mathbb{E}B_1^a}{\mathbb{E}(T_1 + T_2)} = \lambda \frac{\mathbf{1}^T D \mathbf{1}}{2}$$

und somit ergibt sich die absolute Kapazität

$$\bar{\lambda}(\delta^{RR}) \; = \; \frac{2}{\mathbf{1}^T D \mathbf{1}} \tag{4.13}$$

Dies ist nun zwar das selbe Ergebnis wie bei δ^{Coin}, jedoch werden wir sehen, dass wesentlich bessere Simulationsergebnisse bei δ^{RR} in Bezug auf die durchschnittlichen Wartezeiten erzielt werden.

4.4 Die Join-the-least-load Strategie

Bei dieser Strategie versuchen wir die Flugzeuge so zu verteilen, sodass in jeder der beiden Warteschlangen ähnlich viele warten. Wir werden also nun versuchen, die uns vorliegenden Daten über die einzelnen Warteschlangen an den Landebahnen zu verwerten um so eine schnellere Abwicklung zu ermöglichen. Ein Zustand auf einer Landebahn $a \in \{I, II\}$ zur Ankunftszeit des n-ten Flugzeuges wäre etwa folgende Situation: Wir betrachten die übrigbleibende Wartezeit (Auslastung) U_n^a und den Typ ξ_n^a des letzten wartenden Flugzeuges auf der Landebahn a, welche das n-te Flugzeug „sieht“, wobei wir dessen Typ und die dadurch entstehende Seperationszeit nicht berücksichtigen.

Ein konkreteres Beispiel: Zum Ankunftszeitpunkt S_n warten m Flugzeuge auf Landebahn I und der letzte Typ sei $\xi_n^I = i_m$. Die Auslastung U_n^I ergibt sich dann aus den Seperationszeiten der wartenden Flugzeuge und der übgrig geliebenen des Flugzeuges, das gerade gelandet ist. Diese kann natürlich auch 0 sein, falls sie gerade zum Zeitpunkt S_n verstrichen ist. Auf der Landebahn II sei die Warteschlange leer und der zuletzt gelandete Typ sei $\xi_n^{II} = i_1$, wodurch sich eine negative Auslastung U_n^{II} ergibt. Man muss nur noch bedenken, dass die Seperationszeit des n-ten Flugzeuges noch nicht bestimmt ist. Wenn das ankommende Flugzeug mit Typ k nun auf Landebahn II gelotst wird und die Seperationszeit $b(j_1, k)$ größer als $-U_n^{II}$ ist, so muss es die Zeit $U_n^{II} + b(j_1, k)$ warten.

Wenn wir diese Situation annehmen und wir davon ausgehen, dass über den Typ des ankommenden Flugzeuges nichts bekannt ist und auf beiden Landebahnen die letzten Typen gleich sind, so ist die optimale Strategie, das n-te Flugzeug zur Landebahn I zu lotsen, falls $U_n^I < U_n^{II}$ und zu II falls $U_n^{II} < U_n^I$.

Wenden wir uns nun der Situation zu, in der wir den Typ des ankommenden Flugzeuges kennen. Wir benötigen folgende Informationen:

$$(u^I, i^I, u^{II}, i^{II}; k)$$

Sei u^a die Auslastung der Landebahn a und i^a sei der Typ des letzten Flugzeuges auf a. k sei der Typ des gerade ankommenden Flugzeuges, für welches nun entschieden werden muss, welche der beiden Landebahnen es anfliegt.

Wir definieren nun die *eigentliche JLL-Strategie* δ^{JLL} durch

$$\delta^{JLL}(u^I, i^I, u^{II}, i^{II}; k) = \begin{cases} I & \text{für } u^I < u^{II} \\ II & \text{für } u^{II} < u^I \end{cases} \tag{4.14}$$

und die *erweiterte JLL-Strategie* δ^{JLL+}, die den Typ des ankommenden Flugzeuges mit einbezieht

$$\delta^{JLL+}(u^I, i^I, u^{II}, i^{II}; k) = \begin{cases} I & \text{für } u^I + b(i^I, k) < u^{II} + b(i^{II}, k) \\ II & \text{für } u^{II} + b(i^{II}, k) < u^I + b(i^I, k) \end{cases} \tag{4.15}$$

In diesem Kapitel haben wir nun diverse Strategien zur Verteilung der Flugzeuge auf zwei Landebahnen kennengelernt. Im Anschluss werden wir unsere Erkenntnisse numerisch testen um herauszufinden, welche Strategie zu bevorzugen ist und ob erste Vermutungen sich bestätigen.

5 Simulation

5.1 Das Programm in Matlab

Die Matlab-Quellcodes für die implementierten Strategien besitzen den folgenden gemeinsamen Definitionsteil. Darin werden die Zwischenankunftszeiten sowie gemäß unserer Annahme der „Traffic Mix“ erzeugt.

```
clear;

% intensity of air traffic
% 3600*lambda = #aircrafts / h
lambda = 0.010;

% number of airplanes
N = 100000;

% separation times
B = [96 120 144; 72 72 96; 72 72 72];

% traffic mix and cumulated traffic mix
TM = [.7 .2 .1];
CTM = cumsum(TM);

% produce vector of airplane types
rand('state',sum(100*clock));

RND = rand(2*N,1);
TYPES = RND(find(RND < 1))(1:N);

for i = 1 : length(CTM)
   TYPES(find(TYPES < CTM(i))) = i;
end;

% produce interarrival times
IAT = exprnd(1/lambda, N, 1);

% obtain arrival times
AT = cumsum(IAT);
```

Listing 1: Erzeugung der stochastischen Größen

Nun zum interessanteren Teil der Implementierung. Es folgen die Algorithmen, welche die Anwendung der einzelnen Strategien umsetzen.

5.1.1 Strategie für eine Landebahn

```
% determine service times
ST(1) = 0;
for i = 1 : N-1
   ST(i+1) = B(TYPES(i), TYPES(i+1));
end;

% calculate waiting times
WT(1) = 0;
for i = 1 : N-1
   WT(i+1) = max(0, WT(i)+ST(i)-IAT(i+1));
end;


% average waiting time in seconds
av_waiting_time = cumsum(WT)(N)/N
```

Listing 2: Auswertung und Ausgabe der Ergebnisse

5.1.2 Strategien für zwei Landebahnen

Die Strategien unterscheiden sich lediglich in der Zuordnung der Flugzeuge auf die beiden Landebahnen. Deshalb werden auch nur diese Quellcodefragmente hier getrennt für einzelne Strategien aufgeführt. Weiter unten findet sich dann ein gemeinsamer Auswertungs- und Ausgabeteil.

```
% assign airplanes to one of the two runways
% 5/7 of heavy ones will be assigned to the first
% other ones will be assigned to the second
T1 = find(TYPES == 1);
T2 = find(TYPES != 1);
RND = rand(2*length(T1),1);
RND = RND(find(RND < 1))(1:length(T1));
R1 = T1(find(RND < 5/7));
R2 = sort([find(RND >= 5/7); T2]);
```

Listing 3: Splitting

Zu obiger Implementierung zur Strategie der zufälligen Aufteilung auf die Landebahnen sei gesagt, dass es natürlich diverse Möglichkeiten dazu gibt. Wir haben uns hier für die Variante entschieden, bei der 5/7 der Flugzeuge vom Typ 1, also 50% des gesamten Verkehrs, auf Landebahn *I* und alle anderen Flugzeuge auf Landebahn *II* geleitet werden. Dies sollte einer optimalen Verteilung sehr nahe kommen.

```
% assign airplanes to one of the two runways
RND = rand(2*N,1);
RND = RND(find(RND < 1))(1:N);
RND(find(RND < 0.5)) = 1;
RND(find(RND < 1.0)) = 2;
R1 = find(RND == 1);
R2 = find(RND == 2);
```

Listing 4: Fair Coin Flipping

```
% assign airplanes to one of the two runways
% by creating matrices with odd and even indices
R = [1:N];
R1 = find(mod(R,2) == 0);
R2 = find(mod(R,2) == 1);
```

Listing 5: Round Robin

```
% determine service times for each of the two runways
ST1(1) = 0;
for i = 1 : length(R1)-1
   ST1(i+1) = B(TYPES(R1(i)), TYPES(R1(i+1)));
end;

ST2(1) = 0;
for i = 1 : length(R2)-1
   ST2(i+1) = B(TYPES(R2(i)), TYPES(R2(i+1)));
end;


% calculate waiting times for each of the two runways
WT1(1) = 0;
for i = 1 : length(R1)-1
   WT1(i+1) = max(0, WT1(i)+ST1(i)-(AT(R1(i+1))-AT(R1(i))));
end;

WT2(1) = 0;
for i = 1 : length(R2)-1
   WT2(i+1) = max(0, WT2(i)+ST2(i)-(AT(R2(i+1))-AT(R2(i))));
end;


t1 = cumsum(WT1)(length(R1))/length(R1);
t2 = cumsum(WT2)(length(R2))/length(R2);

% average waiting time in seconds
av_waiting_time = ( t1 + t2 ) / 2
```

Listing 6: Auswertung und Ausgabe der Ergebnisse

5.2 Ein numerisches Beispiel

Um einen Einblick zu bekommen, wie die verschiedenen Strategien aus dem vorigen Kapitel in der Praxis funktionieren, werden wir ihre Leistungsfähigkeit an einem Beispiel testen. Wir nehmen dafür an, dass es drei verschiedene Typen (schwer, mittel, leicht) von Flugzeugen gibt. Die Matrix $(b(i,j))_{i,j\in J}$ sei etwa gegeben durch

$$\begin{pmatrix} 96 & 120 & 144 \\ 72 & 72 & 96 \\ 72 & 72 & 72 \end{pmatrix}$$

und der „Traffic Mix“ $p := (0.7, 0.2, 0.1)$. Die durchschnittliche Bedienzeit beträgt nun $\mathbf{1}^T D\mathbf{1} = 96$.

Absolute Kapazität

Die Strategie *Fair Coin Flipping* hat eine absolute Kapazität von

$$\bar{\lambda}(\delta^{Coin}) = \frac{2}{\mathbf{1}^T D\mathbf{1}} = \frac{2}{96} = 0.02083 \quad [Flugzeuge/sek]$$

und ergibt somit eine Kapazität von 75 Flugzeugen pro Stunde.

Ergebnisse der Simulation

Wir vergleichen die Leistungsfähigkeiten verschiedener Strategien durch Berechnung ihrer durchschnittlichen Wartezeiten. Für jedes λ haben wir den Durchschnitt von 100.000 zufälligen Flugzeugen simuliert. Die Ergebnisse kann man der Tabelle 1 entnehmen.

λ	#aircrafts/h	one_runway.m	splitting.m	fair_coin_flipping.m	round_robin.m
0.010	36.0	1262.30	54.97	45.65	16.39
0.011	39.6	2.69e+05	65.46	54.97	21.52
0.012	43.2	6.18e+05	78.17	67.29	26.08
0.013	46.8	9.21e+05	93.08	82.90	33.66
0.014	50.4	1.25e+06	116.38	100.61	40.91
0.015	54.0	1.46e+06	148.26	129.75	52.78
0.016	57.6	1.66e+06	196.50	158.35	70.72
0.017	61.2	1.86e+06	271.72	213.17	104.93
0.018	64.8	2.02e+06	512.92	289.22	148.86
0.019	69.4	2.16e+06	1521.70	491.64	243.26
0.020	72.0	2.32e+06	2.72e+04	1180.20	549.91

Tabelle 1: Ergebnisse der Simulation

6 Konklusion

Unsere anfänglichen Vorstellungen des Modells wurden während der Bearbeitung wesentlich verändert. Wir sind von der Idee mit den unterschiedlichen Schleifen und deren Größen abgekommen. Diese Überlegungen haben wir fallen gelassen durch die Annahme, dass die Flugzeuge jederzeit ihre Geschwindigkeit ändern und so immer zum ehest möglichen Zeitpunkt ihren Landeanflug beginnen können. Stattdessen haben wir verschiedene Strategien betrachtet, wie man Flugzeuge auf zwei Landebahnen verteilen könnte. Bei dem Modell mit einer Landebahn sind wir nach dem FIFO-Prinzip vorgegangen. Dieses Modell könnte man komplexifizieren, indem man die Treibstoffkapazitäten zusätzlich berücksichtigt, was uns dann auf das PRI-Prinzip führen würde. Diese zusätzliche Komponente könnte man natürlich auch im Modell mit zwei Landebahnen mit einbeziehen. Störungen, wie schlechte Wetterverhältnise, und deren Auswirkungen auf Stabilität und durchschnittliche Wartezeiten könnte man in einer erweiterten Ausführung untersuchen. Wir haben außerdem, wie anfangs erwähnt, die Flugzeit vom Eintritt in den Raum der Flugsicherung bis zur tatsächlichen Landung nicht mit einberechnet.

Das Modell mit einer Landebahn ist äußerst instabil, sobald die Rate der ankommenden Flugzeuge den Wert von 37.5 pro Stunde überschreitet. Die Wartezeiten explodieren ab diesem Schwellwert. Bei der Analyse der Strategien für zwei Landebahnen bestätigten sich in den Simulationen die theoretischen Ergebnisse. Die einzige Überraschung ist die starke Überlegenheit von Round Robin gegenüber Fair Coin Flipping trotz der gleichen absoluten Kapazitäten. Leider konnten wir auf Grund von Implemetierungsschwierigkeiten die Strategie JLL nicht numerisch vergleichen. Diese dürfte jedoch mit Abstand die beste sein.

Literatur

[1] John A. Sokolowski, Catherine M. Banks (2009) *Principles of Modeling and Simulation*, John Wiley & Sons, Inc.

[2] Vladimir V. Kalashnikov (1994) *Mathematical Methods in Queuing Theory*, Kluwer Academic Publishers

[3] François Baccelli, Pierre Brémaud (2003) *Elements of Queueing Theory*, Springer-Verlag Berlin Heidelberg New York