Discrete Event Systems
Exercise 12

1 Competitive Analysis

In this exercise, we analyze algorithms for cellular networks such as GSM. In such networks, the area is segmented into cells, each of which containing a base station. Due to interference, base-stations in neighboring (adjacent) cells cannot use the same carrier frequencies, but frequencies may be reused in non-interfering cells (i.e., cells that are not neighboring). In this exercise, we use the idealized hexagonal grid for modelling these cells (cf Figure 1).

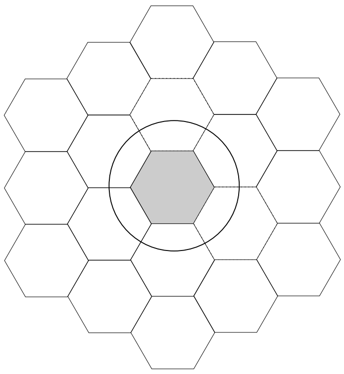

Figure 1: The hexagonal grid modelling the cells.

The number of frequencies in real GSM networks is limited. If there are more callers than channels, some calls must be rejected. In this exercise, we make the simplifying assumption that there is only a *single channel* shared by all base stations. That is, every base station can accept exactly one call. Furthermore, if a a call is accepted by a base station in a certain cell, no calls can be accepted in neighboring cells due to interference.

Assume that callers arrive in online fashion, that is, one after another in an input sequence σ. We need to accept or reject callers such that there is at most 1 caller in a cell and its 6 neighboring cells. The *benefit* of the algorithm is the number of calls we accept.

a) Assume that calls have *infinite duration*. Once a call is accepted, it remains forever. Describe a natural greedy algorithm for the problem. Is your algorithm ρ-competitive for any fixed constant ρ? If so, what is the value of ρ?

b) Assume that every call can have an arbitrary duration, but base stations are not allowed to preempt accepted calls. Propose a competitive online algorithm for this scenario.

c) Let us return to the case of calls with infinite duration. Do you think that your algorithm in a) is optimal? Is it possible to come up with a better online algorithm? Explain your decision.

2 Online Algorithm [last year's exam problem]

Mario und Luigi sind Brüder und arbeiten als Eiscremeverkäufer an einem ein Kilometer langen Strandabschnitt an der Adria.[1] Immer wenn ein Badetourist ein Eis kaufen möchte (Request), muss mindestens einer der beiden zu dem Touristen eilen, und ihm ein Eis verkaufen. Da der Strand sehr flach und übersichtlich ist, können Mario und Luigi immer den ganzen Strand überblicken. Das heisst, bei einem neuen Request kennen Mario und Luigi immer sofort den Ort des Requests. Requests kommen nacheinander und die Zeit zwischen zwei aufeinanderfolgenden Requests ist immer genug lange, damit der Weg zum Kunden zurückgelegt kann, bevor der nächste Request gestellt wird. Wir nehmen also an, dass es nie zwei unerfüllte Requests gibt.

In Bild 2 ist eine Beispielsituation dargestellt. Der $i$te Request befindet sich rechts von Luigi. Im Beispiel wird dieser Request von Luigi bedient. Dieser legt dafür einen Weg von 0.2 Kilometern zurück. Der $i+1$te Request liegt zwischen Luigi und Mario. Wer soll ihn bedienen?

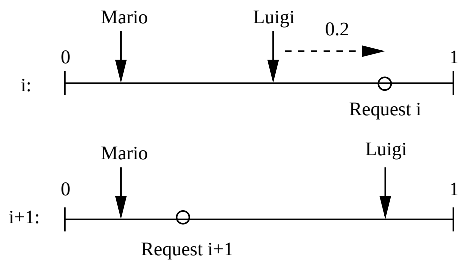

Figure 2: Der Strand mit Mario und Luigi's Position. Der $i$te Request wird von Luigi bedient.

Nehmen Sie an, dass Mario am Anfang auf Position 0 und Luigi auf Position 1 stationiert sind. Ein Request i ist definiert durch seine Position r_i im Interval $[0, 1]$. Mario und Luigi werden konfrontiert mit einer Sequenz von Requests $\sigma = r_1, r_2, \ldots$.

Sei d_M resp. d_L der gesamte Weg, den Mario resp. Luigi an einem Tag (es gibt potentiell unendlich viele Requests) zurücklegen. Um am Abend nicht müde zu sein, möchten Mario und Luigi den Weg, den sie zusammen zurücklegen so kurz halten wie möglich. Das heisst, sie möchten den Wert $d_M + d_L$ minimieren. Mit welcher Strategie sollten Mario und Luigi die Requests bedienen?

a) Mario und Luigi lassen sich von der renommierten Consulting-Firma Cons-ULT beraten. Diese schlagen vor, dass immer derjenige den Request i bedienen soll, der momentan näher bei r_i ist. Der andere soll einfach an seinem gegenwärtigen Ort bleiben und warten. Gemäss Cons-ULT ist dies die bestmögliche Strategie für Mario und Luigi.

Verifizieren Sie die Aussage von Cons-ULT anhand folgender Request Sequenz σ_1:

$$\sigma_1 = 0.4\ ,\ 0.1\ ,\ 0.5\ ,\ 0.7\ ,\ 0.0$$

Beschreiben Sie anhand von Skizzen den Weg den Mario und Luigi zurücklegen. Was wäre die optimale Lösung für Mario und Luigi gewesen? Wie kompetitiv ist der Cons-ULT-Algorithmus bezüglich der Sequenz σ_1?

b) Ist die Strategie von Cons-ULT c-kompetitiv für irgendeine Konstante c? Wenn ja, für welche Konstante? Beweisen Sie ihre Antwort.

c) Es gibt einen 2-kompetitiven Algorithmus ALG für das Problem von Mario und Luigi. Wie der Cons-ULT Algorithmus ist auch ALG deterministisch, also nicht randomisiert. Wie könnte dieser 2-kompetitive Algorithmus aussehen? Schlagen Sie einen deterministischen Algorithmus vor und begründen Sie, warum dieser 2-kompetitiv ist.

[1] Unfortunately, Mario and Luigi cannot speak English fluently and we therefore deem it appropriate to pose this exercise in German.