

IASHeldenVormBildschirm - PharmaBus

David Klein, Manuel Karl, Robina Meyer, Simon Koch

March 22, 2020

1 Vorstellung der Beteiligten

2 Problemstellung und Ziele

Das Problem der Medizinischen Versorgung der Bevölkerung, insb. von Risikogruppen die besonders auf diese angewiesen sind, ist ein facettenreiches Problem. Hierbei stellt sich nicht nur die Frage der konkreten Behandlung sondern auch, wie man eine Behandlung versorgt, i.e., wie können wir sicherstellen, dass die benötigten Medikamente auch beim Patienten ankommen.

Dieses Problem verschärft sich noch einmal in den Zeiten einer globalen Pandemie wie sie COVID-19 darstellt. Zu Zeiten einer Pandemie gilt es zu vermeiden, dass Risikogruppen sich unnötig außerhalb und in Nähe anderer, potentiell infektiöser, Mitmenschen aufhalten. Genau dies ist aber der Fall wenn ein Patient sich Medikamente von der Apotheke abholen muss um seine Behandlung in den eigenen vier Wänden fortführen zu können.

Dieses vermeidbare Problem wollen wir angehen und eine Lösung präsentieren die es erlaubt Patienten zu Hause mit den Medikamenten zu versorgen die sie benötigen. Desweiteren soll die Lösung für eine hohe Skalierbarkeit und deutschlandweites Deployment ausgelegt sein.

2.1 Ziele

Zusammenfassend können wir also sagen, dass die Lösung die wir anstreben die folgenden Ziele hat:

- Alle Patienten sollen mit den benötigten Medikamenten versorgt werden
- Fahrer sollen so sparsam (wenig Fahrer) und so effektiv (ein Fahrer bedient max. Anzahl an Patienten) genutzt werden wie möglich
- Eine zentrale Webseite soll die Aufnahm und Verteilung von Bestellungen regeln, sodass man Kapazitäten von Freiwilligen und Apotheken aggregieren kann

2.2 Out of Scope

Da wir im Rahmen des Hackathon nur begrenzt Zeit hatten und eine allumfassende Lösung mehr Manntage erfordert hätte und uns für andere Aspekte das Wissen fehlte sind die folgenden Punkte für uns out of scope:

- Apotheken wollen möglicherweise nicht ihr Lager leer verkaufen um Laufkundschaft bedienen zu können und eine Verteilung von Bestellungen basierend auf Lagerbeständen würde dies ermöglichen
- Die Routenfindung für einen Fahrer der mehrere Apotheken/Patienten bedienen soll ist komplex und eine optimale Lösung wäre eine Formulierung des 'Travelling Salesman Problem' und somit NP-Vollständig
- Es besteht die Möglichkeit das zu wenig Medikamente für alle Bestellungen verfügbar sind. Dies ist ein ethisches Problem.
- Wir sind nicht bestrebt Ressourcen schonend die beste Kombination zu finden um die insgesamte Fahrstrecke aller Fahrer zu minimieren
- Manche Medikamente sind rezeptpflichtig und eine Übertragung des Rezept muss möglicherweise direkt vom Arzt an die Apotheke gehen, wenn der Patient das Rezept nicht zu Hause vorliegen hat. Dies ist ein rechtliches Problem.

2.3 Mögliche Verbesserungen

Es gibt ein paar Features die wir initial gerne Implementiert hätten, für die uns aber am Ende die Zeit fehlte. Die Grundstruktur dafür ist aber im Prototypen bereits vorhanden, sodass diese Features in wenigen Mannstunden nachgerüstet werden können ohne das Gesamtkonzept überdenken zu müssen:

- Fahrdistanz eines Fahrer kann limitiert werden um einen Fahrer nicht endlos lange Strecken fahren zu lassen
- Ein Fahrer kann möglicherweise nur begrenzt viel gekühlte, oder allgemeine Ladung transportieren
- Benachrichtung der Fahrer über Streckenzuweisung via Email (oder jedwedem anderen notification Dienst)
- Die Eingabemaske und Prozess für die Bestellungen ist in dieser Form vermutlich nicht ausreichend da hier spezifische Produktkennzeichnungen weggelassen wurden die zur Identifikation der Medikamente notwendig wären

3 Heuristic

Im folgenden noch die kurze Definition unserer derzeit eingesetzten Heuristik die probiert

- möglichst wenig Fahrer einzusetzen
- möglichst kürzeste Route zu finden

hierbei wird aber gerade der letzte Punkt sehr heuristisch gelöst und es kann schnell zu Sidecases kommen, indem die Route nicht optimal ist. Die Heuristik ist programmintern aber ausreichend wegabstrahiert, dass ein einfaches austauschen trivial ist.

4 Input

$$\begin{aligned} D &= \{\text{set of driver}\} \\ d_d : D &\rightarrow \mathbb{N} \text{ (max distance of driver)} \\ A &= \{\text{set of pharmacies}\} \\ P &= \{\text{set of patients}\} \\ M &= \{\text{set of meds}\} \\ d_{DA} : D \times A &\rightarrow \mathbb{N} \text{ (distance between driver and pharmacy)} \\ T_{DA} &= \{(d, a) \in D \times A \mid d_{DA}((d, a)) \leq d_d(d)\} \\ T_{DP} &= \{(d, p) \in D \times P \mid d_{DP}((d, p)) \leq d_d(d)\} \\ N &= \{\text{set of needed meds by patients}\} \subset P \times M \\ S &= \{\text{set of stored meds by pharmacies}\} \subset A \times M \\ PD &= \{(d, a, p, m) \in D \times A \times P \times P \times M \mid \\ &\quad (d, a) \in T_{DA} \wedge (d, p) \in T_{DP} \wedge (a, m) \in S \wedge (p, m) \in N\} \end{aligned}$$

5 Algorithmus

Zuerst müssen wir berechnen welche Schritte ein Fahrer abarbeiten soll, bevor wir die optimale route konzipieren

Der vorherige algorithmus hat uns nun ein set gegeben bei dem alle patienten bedient werden und zwar von der minimalen anzahl an fahrern. Wir betrachten nun einen algorithmus der für ein set das alle tuple beinhaltet die einem einzelnden fahrer zugeordnet sind eine optimale route berechnet.

```

sort  $PD$  by driver , pharmacy ,  $p$  ,  $m$ 
drives = {}
do until  $PD$  is empty
     $tuple \leftarrow first(PD)$ 
    drives  $\leftarrow tuple$ 
    delete in  $PD$  where element  $(-, -, p, m)$ 
    reduce  $m$  in stock of  $p$ 
    if stock of  $m$  in  $p$  empty
        delete in  $PD$  where element  $(-, -, p, -)$ 
return drives

```

Figure 1: Dieser Algorithmus gibt uns eine heuristisch optimierte zuteilung von fahrern zu routen von apotheken zu patienten mit medikament

```

input set  $D \subset PD$ 
drive_order = {}
sort  $D$  by frequency( $a$ )
 $a = first(D)[1]$ 
driver_order  $\leftarrow a$ 
rem_steps = {  $p$  in  $D$  who were served by  $a$  and all remaining  $a$  in  $D$  }
// quadratic solution of TSP
do until rem_steps is empty
    closest  $\leftarrow smallest\_distance(drive\_order[-1], rem\_steps)$ 
    drive_order  $\leftarrow closest$ 
    delete from rem_steps closest
    if closest is pharmacy
        rem_steps += { $p$  in  $D$  who were served by closest}

```

Figure 2: Dieser Algorithmus gibt uns die 'optimale route' für einen fahrer