

Detailbeschreibung zum

2. CAU-FLS-Programmierwettbewerb

1 Problembeschreibung

Das Problem ist ein Planungsproblem für einen Pizzalieferservice. Es besteht darin, zu entscheiden

- wieviele Fahrer und Fahrzeuge in einer Schicht eingesetzt werden sollen,
- welche Bestellungen angenommen werden sollen und
- wann und in welcher Reihenfolge die Aufträge geliefert werden (Routenplanung).

Es handelt sich um ein Online-Problem mit einer Vorbereitungsphase.

Vorbereitungsphase Das Programm erhält Informationen zum Zeitraum und der zu erwartenden Anzahl Bestellungen. Es muss festlegen, wieviele Fahrzeuge eingesetzt werden sollen.

Hauptphase Das Programm erhält in Echtzeit Bestellungen. Es muss sofort entschieden werden, ob die Bestellung angenommen wird. Die Disposition der angenommenen Aufträge erfolgt zu beliebigen Zeitpunkten in Form von Touren für die Fahrzeuge.

Ziel ist es, einen möglichst hohen Profit zu erzielen. Dabei werden verschiedene Faktoren berücksichtigt:

- Jedem in der Vorbereitungsphase angeforderten Fahrer muss sein Stundensatz für die gesamte Schicht gezahlt werden, unabhängig davon, ob und wie viel er eingesetzt wird.
- Jeder gefahrene Kilometer kostet einen festen Betrag.
- Die Produktion der bestellten Ware kostet Materialkosten, die von der Bestellung abhängen. Jeder angenommene Auftrag wird automatisch sofort produziert, unabhängig vom Auslieferungszeitpunkt.
- Für jede ausgelieferte Bestellung wird der Bestellwert eingenommen, es sei denn, der vereinbarte späteste Lieferzeitpunkt wurde überschritten.

Wir nehmen vereinfachend an, dass es keine Kapazitätsbeschränkungen gibt:

- Die Produktionszeit ist unabhängig davon, wie viele andere Aufträge gerade bearbeitet werden.
- Fahrzeuge können beliebig viel zuladen.
- Das Programm kann beliebig viele Fahrzeuge und Fahrer anfordern.

2 Modellierung

Allgemein bezeichnen wir immer Zeitpunkte mit t , die Dauer zwischen zwei Zeitpunkten mit δ und Kosten mit c . Es ergeben sich folgende Bezeichnungen:

- n : Anzahl der möglichen Kunden.

- $\delta(i, j)$ für $i, j \in \{0, \dots, n\}$: Fahrzeit zwischen Kunde i und Kunde j in Minuten. Die Werte $\delta(0, i)$ und $\delta(i, 0)$ geben die Fahrzeiten zwischen dem Kunden i und dem Pizzaservice an.
- $c(i, j)$ für $i, j \in \{0, \dots, n\}$: Fahrkosten für die Fahrt von Kunde i zu Kunde j in €. Die Werte $c(0, i)$ und $c(i, 0)$ geben die Kosten für die Fahrt zwischen dem Kunden i und dem Pizzaservice an.
- c_t : Stundensatz der Fahrer in €/h
- δ_ℓ und δ_d : Be- und Enladezeiten der Fahrzeuge in Minuten.
- t_s und t_e : Zeitpunkt des Schichtbeginns und des Schichtendes
- k : Anzahl der Bestellungen
- d : Anzahl der angeforderten Fahrer/Fahrzeuge

Die Werte t_s , t_e und k sind Teil der Eingabe, der Wert d wird in der Vorbereitungsphase vom Lösungsprogramm gewählt. Alle anderen Werte sind konstant und werden rechtzeitig bekannt gegeben. Für jede Bestellung o legen wir ferner fest:

- $i(o)$: Kunden-Index des Bestellers
- $t_o(o)$: Bestellzeitpunkt
- $t_\ell(o)$: Spätester Lieferzeitpunkt
- $\delta_p(o)$: Produktionsdauer in Minuten
- $c_p(o)$: Produktionskosten in €
- $v(o)$: Bestellwert in €

Die Fixkosten sind demnach $F = d \cdot (t_e - t_s) \cdot c_t$. Seien O alle Bestellungen, $O_a \subseteq O$ alle angenommenen Bestellungen. Die Produktionskosten betragen dann $P = \sum_{o \in O_a} c_p(o)$.

Eine Liste $T = (o_1, \dots, o_m)$ von Bestellungen bezeichnen wir als Tour. Sei t_d der Zeitpunkt, zu dem T disponiert wird. Die Tour kann erst begonnen werden, wenn die Ware produziert und verladen wurde. Sei t_v der Zeitpunkt, zu dem das nächste Fahrzeug zur Verfügung steht, bzw. $t_v = t_d$ wenn bereits ein Fahrzeug wartet. Die Fahrzeuge werden in Reihenfolge der Disposition reserviert. Dann wird T zum Zeitpunkt

$$t_0 = \delta_\ell + \max(\{t_o(o_r) + \delta_p(o_r) \mid 1 \leq r \leq m\} \cup \{t_v, t_d\})$$

gestartet. Die Ware erreicht den ersten Kunden zum Zeitpunkt $t_1 = t_0 + \delta(0, i(o_1)) + \delta_d$. Für $2 \leq r \leq m$ erhält der r -te Kunde seine Ware zum Zeitpunkt

$$\begin{aligned} t_r &= t_{r-1} + \delta(i(o_{r-1}), i(o_r)) + \delta_d \\ &= t_0 + r \cdot \delta_d + \delta(0, i(o_1)) + \sum_{q=2}^r \delta(i(o_{q-1}), i(o_q)). \end{aligned}$$

Das Fahrzeug erreicht schließlich zum Zeitpunkt $t_{m+1} = t_m + \delta(i(o_m), 0)$ wieder den Pizzaservice.

Die Lieferung von o_r ist pünktlich, falls $t_r \leq t_\ell(o_r)$ ist. Für jede pünktliche Lieferung o wird $v(o)$ eingenommen, für jede unpünktliche Lieferung werden dagegen Kosten in Höhe von $v(o)$ berechnet (Gutschein). Seien $O(T) = \{o_1, \dots, o_m\}$ alle Bestellungen und $O_p(T) \subseteq O_a$ die pünktlich gelieferten Bestellungen von T . Die Kosten der Tour T betragen dann $C(T) = c(0, i(o_1)) + \sum_{q=2}^m c(i(o_{q-1}), i(o_q)) + c(i(o_m), 0) + \sum_{o \in O(T) \setminus O_p(T)} v(o)$. Der Erlös durch T beträgt schließlich $E(T) = \sum_{o \in O_p} v(o)$.

Seien T_1, \dots, T_q alle disponierten Touren. Zu maximieren ist der Gesamtgewinn $\sum_{j=1}^q (E(T) - C(T)) - F - P$.

3 Daten

Die Be- und Entladezeiten sind durch $\delta_e = 1$ min und $\delta_d = 2$ min gegeben. Die Anzahl n der Kunden und die Kosten- und Fahrzeitmatrizen sind in einer Datei gegeben. Das Dateiformat enthält n in der ersten Zeile. In den folgenden $n + 1$ Zeilen stehen die Fahrzeiten, die i -te dieser Zeilen enthält dabei die Fahrzeit in Minuten von Kunde $i - 1$ zu allen Kunden und dem Lieferservice. Darauf folgen $n + 1$ Zeilen, die die Fahrkosten in der gleichen Reihenfolge angeben. Die Werte $c(i, i)$ und $\delta(i, i)$ sind jeweils explizit mit 0 angegeben. Es folgt ein Beispiel mit $n = 3$.

```

3\n
0 10 3 5\n
11 0 7 13\n
3 8 0 6\n
5 13 6 0\n
0 3.46 0.95 1.44\n
3.46 0 2.02 4.10\n
0.95 2.29 0 1.80\n
1.56 3.99 1.80 0\n

```

Hier dauert beispielsweise die Fahrt vom Lieferservice zu Kunde 1 genau $\delta(0, 1) = 10$ min und die Rückfahrt $\delta(1, 0) = 11$ min, die Fahrt von Kunde 2 zu Kunde 1 dauert $\delta(2, 1) = 8$ min, und die Fahrt von Kunde 3 zu Kunde 1 kostet $c(3, 1) = 3,99$.

4 Auswertung

Die Auswertung erfolgt automatisch durch ein Auswertungsprogramm, das die Bestellungen in einer Zeitperiode simuliert. Die Bestellzeiten dieser Bestellungen liegen zwischen t_s und t_e . Die Programme aller Wettbewerbsteilnehmer werden mit denselben Bestellungen getestet.

4.1 Simulation

Die Simulation erfolgt nicht in Echtzeit, sondern um den Faktor 12 skaliert, d. h. eine Stunde simulierte Zeit benötigt 5 Minuten Realzeit. Die Simulation startet mit dem Zeitpunkt t_s . Sie endet, sobald

1. es nach t_e keine Bestellungen mehr gibt, die angenommen aber noch nicht ausgeliefert sind,
2. nach t_e fünf simulierte Minuten lang (entspricht 25 Sekunden Realzeit) keine Tour disponiert wird, obwohl es noch angenommene, nicht disponierte Bestellungen und ein bereitstehendes Fahrzeug gibt, oder
3. eine Bestellung nicht innerhalb von einer simulierten Minute (entspricht 5 Sekunden Realzeit) angenommen oder abgelehnt wurde.

Wird mit Bedingung 2. beendet, so werden für die nicht ausgelieferten Bestellungen zusätzlich zu den Produktionskosten der Bestellwert als Umsatzreduktion gerechnet (Gutschein). Beendigung der Simulation durch Kriterium 3. führt zur Disqualifikation des Programms. Es ist zu beachten, dass durch dieses Kriterium eine frühzeitige Beendigung des Teilnehmerprogramms eine Disqualifikation zur Folge haben kann.

4.2 Schnittstelle zwischen Teilnehmer- und Auswertungsprogramm

Die Kommunikation zwischen Teilnehmerprogramm und Auswertungsprogramm erfolgt zeilenweise über die Standard-Ein- und Ausgabeströme der Teilnehmerprogramme. Jede Ein- und Ausgabe wird mit einem Zeilenumbruch `\n` abgeschlossen.

Vorbereitungsphase

In der Vorbereitungsphase schreibt das Auswertungsprogramm in jeweils einer Zeile die Startzeit t_s und die Endzeit t_e im Format hh:ss sowie die Anzahl k der zu erwartenden Bestellungen in die Standard-Eingabe des Teilnehmerprogramms. Eine Beispieleingabe ist

```
11:00:00\n
15:30:00\n
100\n
```

Es erwartet, dass das Teilnehmerprogramm eine einzelne Zeile in seine Standard-Ausgabe schreibt, die die Anzahl der verwendeten Fahrzeuge enthält. Eine Beispielausgabe ist `5\n`. Sobald diese Ausgabe erfolgt ist, beginnt die Simulation.

Hauptphase

Die Simulation beginnt mit dem Zeitpunkt t_s . Seien o_1, \dots, o_k die Bestellungen sortiert nach der Bestellzeit, also $t_o(o_1) \leq \dots \leq t_o(o_k)$. Dann wird, sobald die simulierte Zeit $t_o(o_r)$ erreicht ist (hierbei ist die skalierte Simulationszeit zu beachten), die Bestellung o_r in die Standardeingabe des Teilnehmerprogramms geschrieben. Verwendet wird dabei folgendes Format:

```
ORDER  $r$   $i(o_r)$   $t_o(o_r)$   $t_\ell(o_r)$   $\delta_p(o_r)$   $c_p(o_r)$   $v(o_r)$ 
```

Geht beispielsweise die dritte Bestellung o_3 um 11:24:17 Uhr ein, in Auftrag gegeben vom Kunden mit der Nummer 499, so könnte die Eingabe so aussehen:

```
ORDER 3 499 11:24:17 12:09:17 12 4.56 11.23\n
```

Danach muss das Teilnehmerprogramm entscheiden, ob diese Bestellung angenommen wird. Dazu sollte es in seine Standardausgabe eines der Kommandos

```
ACCEPT  $r$ 
REJECT  $r$ 
```

schreiben, in obigem Beispiel also `ACCEPT 3\n` oder `REJECT 3\n`. Erfolgt innerhalb von einer simulierten Minute (also 5 Sekunden Realzeit) keine dieser Ausgaben, wird das Programm disqualifiziert. Wird eine ungültige Bestellnummer r (z. B. von einer bereits angenommenen/abgelehnten oder noch nicht eingegangenen Bestellung) angegeben, so wird der Befehl ignoriert.

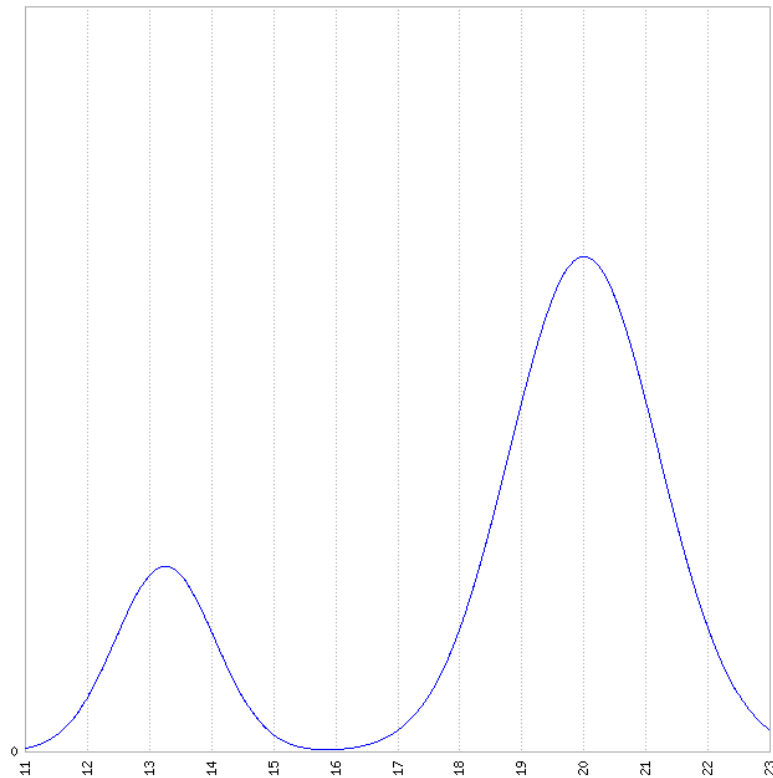
Zu jedem beliebigen Zeitpunkt kann das Teilnehmerprogramm ferner eine Fahrzeugroute angeben. Dazu kann es den Befehl

ROUTE $r_1 \dots r_m$

verwenden, wobei r_1, \dots, r_m die Indizes von angenommenen und noch nicht in einer Route angegebenen Bestellungen sein müssen. Die Route wird zum frühestmöglichen Zeitpunkt (wie in Abschnitt 2 beschrieben) gestartet. Zulässige Routenangaben könnten etwa ROUTE 3\n oder ROUTE 5 2 3\n sein. Enthält eine Tour ungültige Bestellnummern (z. B. abgelehnte oder bereits disponierte Bestellungen), so werden diese in der Tour ignoriert.

4.3 Verteilung der Bestellungen

Die Bestellungen werden zufällig generiert. Die Wahl der Kunden erfolgt gemäß einer Gleichverteilung, wobei kein Kunde mehrere Bestellungen macht. Die Verteilung der Bestellzeiten folgt der Summe zweier Normalverteilungen:



Die zugehörige Verteilungsfunktion ist

$$\frac{4}{5} \cdot \mathcal{N}(20, 1.2) + \frac{1}{5} \cdot \mathcal{N}(13.25, 0.8),$$

wobei $\mathcal{N}(\mu, \sigma)$ die Standard-Normalverteilung bezeichnet.