# Improving Search toepassen op het Running Dinner Probleem

Bram Schellekens (4850785)
Natasja Crombach (4241991)

Toegepaste Wiskunde
Operations Research 5
Fontys Hogescholen

Oktober 2023

# 1 TABLE OF CONTENTS

2 Probleemdefinitie ..... 3
2.1 Verzamelingen en indices ..... 3
2.2 Parameters ..... 3
2.3 Beslisvariabelen ..... 3
2.4 Probleemeisen ..... 3
2.5 Eisen (hard constraints) ..... 4
2.6 Wensen (soft constraints) ..... 4
3 Verantwoording heuristiek ..... 4
4 Verantwoording buurruimtestructuur ..... 5
5 Pseudocode heuristiek ..... 5
6 Tijdscomplexiteit heuristiek ..... 5
7 Testplan en testresultaten ..... 6
7.1 Testplan ..... 6
7.2 Rekentijd ..... 6
Bibliografie ..... 6

# 2 PROBLEEMDEFINITIE

Roy Willemen heeft de probleemdefinitie opgesteld om voor dit verslag te gebruiken. De omschrijving van de wensen is samengevat, de rest van het probleem is overgenomen van het bestand van Roy. (Roy Willemen)

## 2.1 VERZAMELINGEN EN INDICES

A ≙ verzameling van alle huisadressen, met indices a, $a_1$, $a_2$ ∈ A.
D ≙ verzameling van alle deelnemers, met indices d, $d_1$, $d_2$ ∈ D.
E ≙ verzameling van deelverzamelingen van enkele deelnemers die het gehele running dinner bij elkaar aan tafel zitten, met indices $e_1$, $e_2$ ∈ E.
Element e ∈ E is een deelverzameling van verzameling D (bijvoorbeeld: $e_1 = \{d_1, d_2\} \subseteq D$).
G ≙ verzameling {'Voor', 'Hoofd', 'Na'}, de gangen die gegeten zullen worden, met index g ∈ G.

## 2.2 PARAMETERS

$k_a$≙ 1 als de deelnemer(s) wonend op huisadres $a \in A$ een gang moet(en) koken en dus niet vrijgesteld zijn van koken, 0 anders.
$l_a$≙ minimaal aantal deelnemers (tafelgenoten) op huisadres $a \in A$.
$u_a$≙ maximaal aantal deelnemers (tafelgenoten) op huisadres $a \in A$.
$h_{a,d}$≙ 1 als deelnemer $d \in D$ op huisadres $a \in A$ woont, 0 anders.
$v_{a,g}$ ≙ 1 als de deelnemer wonend op huisadres $a \in A$ een voorkeur heeft voor gang $g \in G$, 0 anders.
$vg_{a,g}$ ≙ 1 als op huisadres $a \in A$ in het vorige jaar gang $g \in G$ werd bereid, 0 anders.
$vt_{d_1,d_2}$ ≙ 1 als twee verschillende deelnemers $d_1, d_2 \in D$ ($d_1 \neq d_2$ ) vorig jaar tafelgenoot waren, 0 anders.
$b_{d_1,d_2}$ ≙ 1 als twee verschillende deelnemers $d_1, d_2 \in D$ ($d_1 \neq d_2$ ) elkaars directe buren zijn, 0 anders.

## 2.3 BESLISVARIABELEN

$x_{a,d,g}$ ≙ 1 als deelnemer $d \in D$ gang $g \in D$ eet op huisadres $a \in A$, 0 anders.
$y_{a,g}$ ≙ 1 als op huisadres $a \in A$ gang $g \in G$ bereid wordt, 0 anders.

## 2.4 PROBLEEMEISEN

$\sum_{a \in A} h_{a,d} = 1$ ∀d ∈ D : Elke deelnemer woont op precies één adres.
$\sum_{g \in G} v_{a,g} \leq 1$ ∀a ∈ $A$ : Elk huisadres heeft hoogstens één voorkeursgang.
$\sum_{g \in G} vg_{a,d} \leq 1$ ∀a ∈ $A$ : Elk huisadres heeft vorig jaar hoogstens één gang bereid.
$l_a \leq u_a$ ∀$a \in A$: Voor elk huisadres zijn $l$ en $u$ correcte onder- en bovengrenzen op het aantal tafelgenoten.

## 2.5 EISEN (HARD CONSTRAINTS)

$\sum_{a \in A} x_{a,d,g} = 1$ ∀d ∈ D ∀g ∈ G : Elke deelnemer eet elke gang precies één keer.

$\sum_{g \in G} y_{a,g} = k_a$ ∀a ∈ A : Elk huishouden moet precies één keer koken.

$x_{a,d,g} \geq y_{a,g} \cdot h_{a,d}$ ∀a ∈ A ∀d ∈ D ∀g ∈ G : Alle koks eten het gerecht dat ze zelf hebben bereid.

$\sum_{d \in D} x_{a,d,g} \geq l_a \cdot y_{a,g}$ ∀a ∈ A ∀g ∈ G : Het aantal gasten mag het maximum niet overschrijden.

$\sum_{d \in D} x_{a,d,g} \leq u_a \cdot y_{a,g}$ ∀a ∈ A ∀g ∈ G : Het aantal gasten mag niet onder het minimum zitten.

$x_{a,d_1,g} = x_{a,d_2,g}$ ∀e $\in E$ $d_1, d_2 \in e$ ∀a ∈ A ∀g ∈ G : Sommige duo's moeten bij elkaar blijven.

## 2.6 WENSEN (SOFT CONSTRAINTS)

$\sum_{a \in A} x_{a_1,d_1,g} + x_{a_1,d_2,g} + x_{a_2,d_1,g} + x_{a_2,d_2,g} \leq 3$ $\forall a_1, a_2 \in A : a_1 \neq a_2$ $\forall d_1, d_2 \in D \setminus E$ : $d_1 \neq d_2$: Er zitten zo min mogelijk deelnemers met meer dan één gang bij elkaar.

$y_{a,\text{'Hoofd'}} \cdot vg_{a,\text{'Hoofd'}} = 0$ ∀a ∈ A : Er zijn zo min mogelijk mensen die twee jaar op rij het hoofdgerecht koken.

$y_{a,g} \geq v_{a,g}$ ∀a ∈ A ∀g ∈ G : Zo min mogelijk koks die niet hun voorkeursgang koken, als ze deze hebben aangegeven.

$(x_{a,d_1,g} + x_{a,d_2,g}) \cdot b_{d_1,d_2} \leq 1$ $\forall a \in A$ $\forall d_1, d_2 \in D$ : $d_1 \neq d_2$ ∀g ∈ G : Zo min mogelijk buren eten samen.

$(x_{a,d_1,g} + x_{a,d_2,g}) \cdot vt_{d_1,d_2} \leq 1 + h_{a,d_1} \cdot h_{a,d_2}$ $\forall a \in A$ $\forall d_1, d_2 \in D$ : $d_1 \neq d_2$ ∀g ∈ G : Zo min mogelijk mensen die vorig jaar samen hebben gegeten, eten nu weer samen.

# 3 VERANTWOORDING HEURISTIEK

Het running dinner probleem wordt beschreven als een "NP-hard" probleem. Dat wil zeggen dat het probleem relatief lastig op te lossen is, ook als dit door een computer wordt gedaan. De moeilijkheidsgraad van het probleem zit voornamelijk in de grote hoeveelheid opties waaruit gekozen kan worden om een oplossing te verkrijgen. Zodra er een oplossing gevonden is, is het daarom efficiënter om deze oplossing aan te passen, in plaats van naar een nieuwe oplossing te zoeken.

Met de discrete improving search methode verandert het gegenereerde rooster telkens een klein beetje. Als deze verandering een verbetering is (en aan de gestelde eisen voldoet), wordt de "tijdelijk optimale oplossing" aangepast. Als dit niet zo is, kijkt het programma verder met de eerder gevonden oplossing. Dit proces wordt herhaald tot er geen verdere verbeteringen te vinden zijn of tot het proces wordt afgekapt. Door de kleine verbeteringen is het een effectieve manier om te zoeken naar een goede oplossing.

# 4 VERANTWOORDING BUURRUIMTESTRUCTUUR

De gekozen buurruimtestructuur bestaat uit de oplossingen waar een voor een (voorgerecht, daarna hoofdgerecht en ten slotte nagerecht) de eetadressen van twee bewoners zijn omgewisseld ten opzichte van het vorige tijdelijke optimum. Bewoner A eet het gewisselde gerecht op het adres waar bewoner B oorspronkelijk zou eten. Zodra berekend is of deze nieuwe variatie klopt, wordt de volgende wissel toegepast.

De gekozen buurruimtestructuur maakt de kans groter dat de nieuwe gevonden oplossing binnen de eisen valt, dan na het wisselen van meerdere bewoners, gangen of andere variabelen per iteratie. Dit betekent dat het programma langzamer draait, in verband met de vele iteraties, maar wel alle beschikbare opties afgaat om een optimale oplossing te zoeken.

Van tevoren aangegeven duo's blijven bij elkaar. Als de een wordt verplaatst, gaat de ander mee en wordt er gecontroleerd of het maximum- en minimumaantal gasten per adres niet overschreden wordt. Duo's die verwisseld worden met een bewoner of met een ander duo horen dus ook bij de buurruimtestructuur.

# 5 PSEUDOCODE HEURISTIEK

1. De ingevoerde planning is het tijdelijke optimum.
2. Zoek naar een nieuw tijdelijk optimum.
    a. Wissel van één gang de eetadressen van twee bewoners om.
    b. Controleer of de oplossing aan alle eisen voldoet.
        i. Ja: Ga door naar 2c.
        ii. Nee: Wissel de adressen terug en ga door naar 2a.
    c. Bereken het aantal keer dat er niet aan de wensen wordt voldaan.
        i. Kleiner dan het aantal keer dat het vorige tijdelijke optimum niet aan de wensen voldoet: Dit is je nieuwe tijdelijke optimum, ga door naar 2a.
        ii. Groter dan het aantal keer dat het vorige tijdelijke optimum niet aan de wensen voldoet: Je tijdelijke optimum blijft hetzelfde, adressen worden terug gewisseld, ga door naar 2a.
3. Herhaal stap 2 totdat alle variaties af zijn gegaan of het programma is stopgezet, ga dan door naar 4.
4. Het huidige tijdelijke optimum is nu het algemene optimum, zet het in een Excel-bestand.

# 6 TIJDSCOMPLEXITEIT HEURISTIEK

De tijdscomplexiteit van de discrete improving search methode is exponentieel. In de grote zoekruimte worden namelijk alle verschillende opties bekeken en er is geen maximum gesteld aan het aantal iteraties.

# 7 TESTPLAN EN TESTRESULTATEN

## 7.1 TESTPLAN

Om de tool te testen, is er gekeken naar de rekentijd die het kost om bepaalde veranderingen aan te brengen en te controleren. Zie onder het kopje "7.2 Rekentijd" de tijd die het zoeken naar een optimale verdeling van de verschillende gangen kost.

Invoeren van verschillende soorten bestanden heeft getest wat er gebeurt als er een bestand met een fout format wordt geüpload: Na het aanpassen van de code achter de interface en het handmatig schrijven van errors, vertelt een error de gebruiker wat er fout gaat.

Met visuele tests is gecontroleerd of het programma de goede stappen doorloopt en daarmee op de gewenste oplossing komt. Denk bijvoorbeeld aan: Het aantal overschreden wensen gaat omlaag of blijft hetzelfde per iteratie, dit resultaat komt overeen met de verwachtingen (door middel van logging). Ook zijn er in het gegenereerde Excel-bestand geen onverwachte dingen te zien.

## 7.2 REKENTIJD

Een tool die de tijd tot op de milliseconde kan meten, maakt bij het starten en het eindigen van het programma een notitie van de huidige tijd en geeft weer hoe lang de rekentijd per gang is. Deze tool liet de volgende tijden zien:

| | |
|---|---|
| Start | 0:00,000 minuten |
| Voor | 2:03,271 minuten |
| Hoofd | 1:12,635 minuten |
| Na | 1:14,900 minuten |
| Totaal | 4:30,866 minuten |

# BIBLIOGRAFIE


ChatGPT. (2023, September). Used for inspiration.

Roy Willemen, F. H. (sd). *Runningdinnerprobleem Vughterpoort 2023 - Math program.pdf*. Opgehaald van Canvas: https://canvas.fontys.nl/courses/16607/files/1946847?module_item_id=556474