Programmeertheorie LEGO

Renee Witsenburg
Johannes Meijer van Putten
Nick De Dycker

May 6, 2014

1 Inleiding

De case vraagt ons om een zo groot mogelijke vloeroppervlakte te maken in LEGO steentjes. Hier gelden wel wat regels. Ten eerste moet de vloeroppervlakte binnen de grens van 25 bij 15 noppen blijven. Echter we mogen wel de hoogte in, tot maximaal 5 verdiepingen. Een verdieping bestaat uit 2 lagen LEGO die even groot zijn qua oppervlakte. Een verdieping hoeft niet even groot te zijn als die daaronder. Hij mag echter niet groter zijn qua oppervlakte dan zijn voorganger. Uiteraard mogen we geen gaten of uitsteeksels hebben in onze vloer. De volgende LEGO steentjes mogen gebruikt worden in deze case:

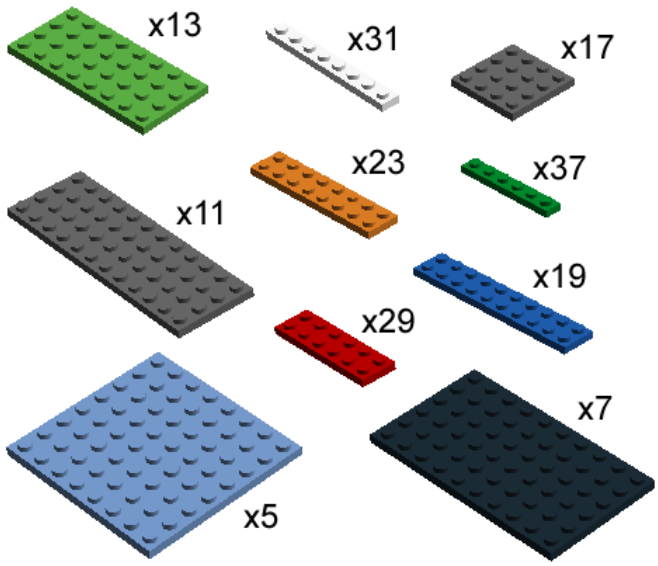

Figure 1: LEGO stenen die gebruikt mogen worden in de case om een gebouw mee op te bouwen.

Eenmaal met de hand een maximaal haalbare oppervlakte gevonden, wordt de case uitgebreid. Het doel wordt uitgebreid door te zoeken hoe toevallig het is dat we met de hand een maximale oppervlakte hebben gevonden. We willen alle mogelijkheden voor 5 dubbelsets genereren. Deze dubbelsets moeten corresponderen met de vijf etages van het tot nu toe bereikte maximum. Bovendien zal dit computerprogramma als ingangspunt voor een eventuele solver.

1.1 Optimum

Er zijn een paar manieren om het optimum te berekenen. De grootste haalbare oppervlakte van een verdieping is 24*15 = 360. 25*15 is namelijk niet haalbaar opdat je dan een lege ruimte overhoud

en dat mag volgens de voorwaarde van de case niet. Als je de ruimte 1x15 zou willen opvullen om 25x15 te maken, blijkt dat dit niet kan met de gegeven LEGO steentjes. Deze blijken allemaal even te zijn. Met 1x8 en 1x6 zouden we het dichtst in de buurt komen, maar maakt de rechthoek dus ongeldig. Verder is 25*14 = 350 en is dus kleiner dan 24*15. Omdat we maximaal 5 verdiepingen mogen vullen is het theoretisch optimum 24*15*5 = 1800. Dit is dus theoretisch de maximale score.

Als we nu naar het aantal LEGO stenen wat beschikbaar is, komen we er gauw achter dat we deze maximale score niet kunnen maken. Daarom moeten we op zoek naar de praktisch maximale score van deze case. Hiervoor nemen we de totale oppervlakte die we met deze set van LEGO stenen kunnen maken, delend door twee, want een verdieping bestaat uit 2 lagen LEGO. Dit is de score 1761.

2 Methodes

Om te onderzoeken hoe toevallig het is dat er handmatig een maximale oppervlakte is gevonden (zichtbaar in figuur 2), delen we het probleem op in deelproblemen. Ten eerste zoeken we hoeveel subsets er van een bepaalde oppervlakte gemaakt kunnen worden. Hiervoor maken we lijsten van het aantal LEGO blokjes en maken we combinaties om op de gezochte oppervlakte te komen. Hier houden we nog geen rekening of we hier uiteindelijk een rechthoek van kunnen maken. We zoeken met een depth first methode recursief naar deze subsets. Eenmaal gevonden zijn er nog twee problemen die opgelost moeten worden. Met het resultaat weten we nog niet of de combinatie van LEGO stenen in een rechthoek past. Daarnaast willen we ook rekening houden met het beperkt aantal blokjes voor het hele gebouw.

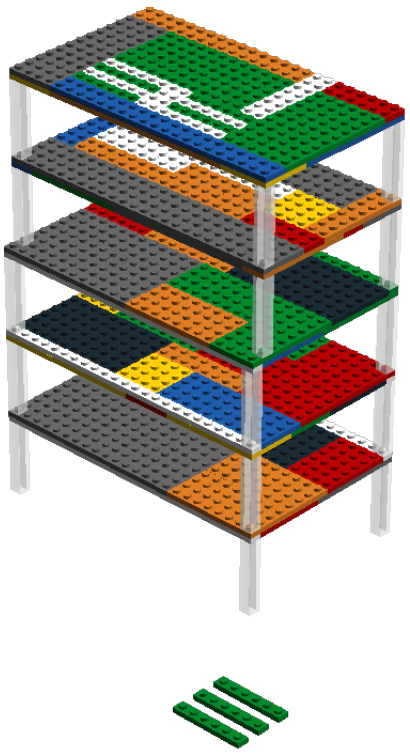

Figure 2: Het handmatige gevonden maximum vloeroppervlakte met een score van 1752.

2.1 Maximale Score Algoritme

De oorsprong van het probleem was het zoeken naar de maximale vloeroppervlakte in LEGO noppen. In de eerste week van de onderzoeksperiode hebben we handmatig vermoedelijk de maximale score gevonden. Om dit hard te maken hebben we uiteindelijk een bewijs hiervoor gevonden.

Om te beginnen hebben we het praktisch optimum van de maximale score, 1761. Een grotere waarde is er niet. Deze score moet verdeeld worden over 5 verdiepingen. Hiermee kunnen we de gemiddelde score per verdieping uitrekenen, in ons geval 352,2. Met deze wetenschap moeten we een rechthoek vinden die groter is dan de gemiddelde score over 5 verdiepingen, maar die wel aan de voorwaarden van de case voldoen. De maximale vloeroppervlakte die we mogen maken is 25x15. Dit kan echter niet omdat we dan een lege ruimte in het vloeroppervlakte overhouden. Daarom kiezen we voor de wel geldige maximale vloeroppervlakte voor de verdieping: 24x15=360. Deze vloeroppervlakte is groter dan het gemiddelde over 5 verdiepingen. Hadden we een vloeroppervlakte van 25x14=350 of een andere maat genomen hadden we niet aan de gestelde voorwaarde voldaan. De gevonden maximale vloeroppervlakte wordt van ons begin optimum afgetrokken. Hieruit berekenen we een nieuw gemiddelde. Opnieuw moeten we een vloeroppervlakte vinden wat voldoet aan de gestelde eisen van de case en wat groter is dan het gestelde gemiddelde van de overige verdiepingen.

Dit algoritme aangehouden, kunnen we bewijzen dat ons handmatig gevonden maximale score, inderdaad met de geldende restricties de maximale score is.

Het blijkt echter ook dat je dit bewijs kunt gebruiken als algoritme bij het zoeken naar andere maximale vloeroppervlaktes. Bij ander opgegeven LEGO stenen en/of andere afmetingen van verdiepingen. pv

Begin praktisch optimum van gebouw		Gemiddeld vloeroppervlakte van 1 verdieping		Gemaakt vloeroppervlakte verdieping		Nieuw optimum	
1761		352,2		24x15=360		1401	
1401		350,25		24x15=360		1041	
1041		347		24x15=360		681	
681		340,5		24x15=360	24x14=336	321	345
321	345	321	345	24x13=312	24x14=336	9	9

Table 1: Maximale Score Algoritme toegepast op onze case

2.2 Depth First Algoritme

Om vanuit het voorgaande algoritme een gebouw te gaan bouwen, moeten we nog de nodige stappen doen. Ten eerste willen we subsets LEGO stenen verzamelen om lagen te maken van de gegeven oppervlakte. Met het depth first algoritme hebben we alle geldige subsets van een laag LEGO stenen, dus een halve verdieping, die de gewenste oppervlakte geven opgezocht en opgeslagen.

Ook het zoeken naar geldige combinaties van subsets om een gebouw mee op te bouwen kan depth first gedaan worden. Hier wordt gezocht naar subsets die samen een nieuwe subset vormen, waarvan er voldoende LEGO stenen beschikbaar zijn om een gebouw met het geldige optimum te bouwen.

2.3 Rechthoeken passen

Vanuit deze opgeslagen subsets van LEGO stenen gaan we op zoek welke subsets geldig zijn om een rechthoek van de juiste afmetingen mee te maken. Een combinatie van 5 stenen van 8x8, een

steen van 8x4 en een blok van 1x8 is samen wel 360, maar zal niet in een rechthoek van 24x15 passen. Daarnaast moeten voor een oppervlakte van 24x15 sowieso een steentje nodig van 1 breed. Dit kan met bijvoorbeeld 4x 1x6 of met 3x 1x8 zijn. Dit zijn methodes om sneller tot een geldige of ongeldige rechthoek te komen.

De methode lijkt een beetje op het bin packing probleem. Een NP probleem voor verschillende toepassingen. Bijvoorbeeld het vullen van containers, vullen van vrachtwagens met een bepaalde ladingscapaciteit of het comprimeren van bestanden op de computer. Wij hebben hier ook een heuristiek gebruikt die bij dit probleem gebruikt wordt. Het first fit algoritme gebruiken we om te zoeken naar een oplossing om een rechthoek te maken. Hoe dit gebeurd maakt niet uit. Zodra er een oplossing is gevonden gaat het programma door met de volgende set LEGO stenen passen. Het verschil is echter wel dat onze sets al op volgorde van groot naar klein staan. Hierdoor werkt het algoritme al beter dan wanneer de stenen niet op volgorde zouden staan.

3 Resultaten

In figuur 2 voor onze met de handgevonden maximale score weergegeven. Dit was het beginpunt om een methode te vinden voor een eventuele solver. Met bovenstaande methodes is ons dat gelukt.

Om te berekenen hoe toevallig het is dat we handmatig een maximale score hebben gebouwd handmatig, hebben we geprobeerd zoveel mogelijk rechthoeken te maken met verschillende LEGO stenen. Om dat compleet uit te rekenen duurde dat te lang. Sommige sets deden er anderhalf uur over om ze te passen. Daarom hebben we voor een aantal resultaten schattingen gemaakt. Hiervoor hebben we van een aantal subsets laten berekenen of ze geldig of ongeldig zijn voor de voorwaarden die we op dat moment stelden.
Voor de 24x15 sets hebben we de computer 30 uur laten rekenen op de eerste 5357 sets. Op dat moment was er nog geen 1% van het geheel berekend. De resultaten tot op dat moment zagen er als volgt uit:
5357 van 571506 sets doorgerekend. Daarvan zijn er 2284, 42.63% geldige rechthoeken. Hieruit hebben we geschat dat er over het geheel dus ongeveer 243633 geldige rechthoeken zullen zijn.

Voor de 24x13 sets hebben we tot nu toe de eerste 2.25% berekend of ze passen. Daarvan past gemiddeld 38% van de sets. Als we hieruit een schatting maken komen we op ongeveer 85460 geldige rechthoeken.

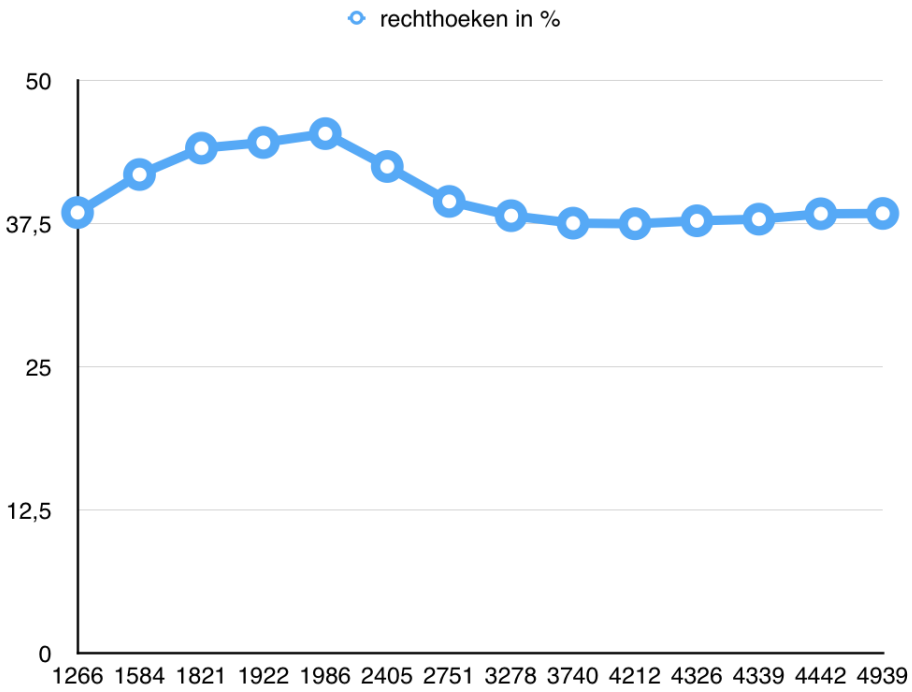

Figure 3: Resultaten van de eerste subsets voor 24x13.

vloeroppervlakte	aantal mogelijkheden (recursies)	geldige mogelijkheden voor de opgegeven oppervlakte
24x15	14523115	571506
24x14	8770189	363170
24x13	5149091	224896

vloeroppervlakte	geldige mogelijkheden voor de opgegeven oppervlakte	geldige rechthoeken (schatting)
24x15	571506	243633
24x14	363170	
24x13	224896	85460

maten gebouw	toestandsruimte	geldige subsets (schatting)
4x 24x15 & 1x 24x13		
3x 24x14 & 2x 24x14		

4 Conclusie

De vraagstelling van deze case is hoe toevallig het is dat wij met de hand een geldig optimum hebben gevonden.
Vanuit de schatting bekeken valt het wel mee hoe toevallig het is dat we handmatig een oplossing hebben gevonden. Er zijn 2 manieren om de maximale score te behalen. Wij hebben maar 1 van die manieren gebruikt om handmatig een oplossing te bouwen. Het feit dat er al 2 manieren zijn

om de maximale score te bouwen, maakt het al iets minder toevallig dat we een maximale score hebben gevonden.

5 Referenties