CHALMERS

Automatisering av pallpackning

Examensarbete inom Data- och Informationsteknik

SIMON RÖHR
VIKTOR HANSSON

Institutionen för Data- och Informationsteknik
CHALMERS TEKNISKA HÖGSKOLA
GÖTEBORGS UNIVERSITET
Göteborg, Sverige 2019

EXAMENSARBETE

Automatisering av pallpackning

SIMON RÖHR
VIKTOR HANSSON

Institutionen för Data- och Informationsteknik
CHALMERS TEKNISKA HÖGSKOLA
GÖTEBORGS UNIVERSITET

Göteborg 2019

Automatisering av pallpackning
SIMON RÖHR
VIKTOR HANSSON

Examinator: JONAS ALMSTRÖM DUREGÅRD

Institutionen för Data- och Informationsteknik
Chalmers Tekniska Högskola / Göteborgs Universitet
412 96 Göteborg
Telefon: 031-772 1000

Institutionen för Data- och Informationsteknik
Göteborg 2019

SAMMANFATTNING

Framtiden går mot ett mer näthandelsbaserat tillstånd och medför ett behov av att effektivt sköta lagring, hantering och leverering av de produkter som på marknaden erbjuds. Robotteknologi är idag bättre och billigare än tidigare, detta till följd av konkurrensen på marknaden. En investering i en automatisering av pakethanteringen är således närmare idag, även för mindre företag som bedriver näthandel och med en lägre budget. Företaget RT Robotics ser en möjlighet att genomföra denna automation och behöver då den algoritm som hanterar och optimerar de paket som är tänkta att placeras på en lastpall.

I denna rapport beskrivs framtagandet av denna algoritm, som med information om paketens dimensioner beräknar fram slutpositioner hos paketen på den tänkta lastpallen. Den kod som skrevs hanterar en stor variation sett till både paketdimensioner och antal mycket väl, något som för en människa kan ses som en näst intill omöjlig uppgift då tiden är begränsande.

Den praktiska utformningen av inläsning- och frammatningsteknik av paketen samt den fysiska placeringen roboten ansvarar för ligger utanför detta projekts ramar och lämnas åt RT Robotics. De paket som hanteras begränsas även till en rätblocksformad geometri, detta för att simplifiera programlogiken.

Nyckelord: effektivisering – automatisering – planering – dimensionering – programmering – reflektion – förbättring

ABSTRACT

The state of commercial trade is getting more web based and with it a need of an effective way of storing, handling and delivering goods. Collaborative robots are today more accessible and smarter than ever before due to the hard competition and the exponential rise in demand. An investment in the automatization of the package handling process are today a serious question, not just for the big global companies but also for those with a much smaller budget. RT Robotics is Gothenburg based automatization company with an urge to fill this need. For them to do so they have requested the algorithm that serves to handle and optimize the palletizing of the final coordinates of the packages.

In this report the making of such an algorithm is explained. With the dimensions of the packages as input, the calculation of an optimized pallet is then calculated. The strength of the final program is the ability to handle large variations in both numbers of packages and different dimensions. A person ability to solve a similar large varying problem will not match that of the program when the time frame is of importance.

The practical implementation of the actual system: scanner, robot and the question of how to feed them both the with packages do not fall within the projects boundaries. In order to simplify the logic of the program, the packages will be restricted to a cuboid geometry.

Keywords: effectivization – automatization – planning – dimensioning – programming – reflection – improvement

FÖRORD

Detta examensarbete är utfört under vårterminen 2019 på Chalmers tekniska högskola. De två eleverna som står för denna publikation studerar högskoleingenjörsutbildningen inom Mekatronik. Vi vill rikta ett stort tack till vår handledare och numera vän, Sakib Sistek. Han har under mörka stunder med hjälp av sin naturliga charm väglett oss genom detta projekt.

Table of Contents

1. INLEDNING

I en värld som för varje dag blir mer och mer globaliserad och där handel länder emellan för varje dag ökar, växer behovet av att på ett så effektivt och smidigt sätt packa, transportera och leverera tillverkade produkter. Säljare till köpare, företag till företag. Varorna i matbutiken är med allt som oftast producerade utomlands, kläderna du bär samt tangentbordet denna inledning skrivs på likaså. Denna livsstil, trots de negativa miljökonsekvenser som uppstår vid transporten av alla dessa produkter, är vår verklighet och smarta lösningar som effektiviserar processen är mycket eftertraktade.

Majoriteten av alla transporter sker idag med hjälp av lastpallar där produkterna läggs i paket och placeras på denna plattform [1]. Lastpallen är utformad så att gaffeltruckar kan hantera varor i större volymer än vad de mänskliga musklerna tillåter. Packningen på lastpallarna sker allt som oftast manuellt, där lagerarbetare utför denna repetitiva och monotona arbetsuppgift. Paketen är med stor sannolikhet rätblocksformade för att kunna staplas på varandra och därmed ge en större packningsvolym på lastpallen [2]. Hos större företag med standardiserade storlekar på paketen kan processen vara automatiserad och robotar sköta arbetsuppgiften. Varför företagen väljer att automatisera packningsprocessen trots den höga grundinvesteringen (robot, transportband, kunskap etc.), är för att det är ekonomiskt fördelaktigt under en längre planeringshorisont. Ett företag behöver inte betala arbetsgivaravgift för en robot, en robot har ingen familj och således inga sjuka barn. Listan kan göras mycket längre men faktum är att en robot kan arbeta kontinuerligt så länge den har en energikälla. Detta är vad som driver övergången till en mer automatiserad produktion- och packningsprocess [3].

Teknik blir med tiden alltid billigare och mer lättåtkomlig. För mindre företag där packningen idag sker manuellt är övergången till en alternativ automatiserad packningsprocess mer genomförbar än innan. Robotar har helt enkelt blivit billigare. En robot behöver dock information och ett bestämt sekventiellt utförandedirektiv. Med andra ord behöver alltså roboten programmeras och en algoritm skrivas, skräddarsydd till den uppgift som roboten är tänkt att utföra.

I detta arbete beskrivs framställningen av ett program där paket scannas in och får en uträknad slutposition på en lastpall. Roboten får med hjälp av detta program information om var paketen ska placeras på lastpallen samt i vilken ordning paketen skall packas. Hur roboten utför placeringen ligger utanför projektets avgränsningar. Projektet beskriver endast hur programmet konstruerats för uträkning av slutgiltiga paketkoordinater på en lastpall.

1.1 Bakgrund

I inledningen av denna rapport förklaras bakgrunden till möjligheten att automatisera packningsprocessen hos mindre företag. RT Robotics är ett nystartat automationsföretag i Göteborg som är nyfikna på möjligheten att utföra denna automatisering. Företaget ser en stor potentiell efterfrågan hos diverse näthandelsbolag som ännu inte har en automatiserad packningsprocess. Roboten som är tänkt att användas är kollaborativ, vilket medför en billigare implementation av denna tänkta övergång. Kollaborativa robotar kännetecknas bland annat av att de är billigare än traditionella industrirobotar [4]. De är enklare att programmera och har inbyggda sensorer som känner av om något eller någon kommer i vägen. Detta gör att en säkerhetszon i tänkt lagerutrymme av säkerhetsskäl ej behöver tas i bejakande. Alltså tar den kollaborativa roboten mycket mindre plats och är därmed bättre anpassad till företag där lagerutrymmet är begränsat.

1.2 Syfte

Uppdraget går ut på att efterforska möjligheterna till att färdigställa ett program som med paketens dimensioner som input (det två första rutorna från vänster i figur 2.1) räknar fram den bästa lösningen att packa paketen på en lastpall (den mellersta gula rutan i figur 2.1). Alltså kommer slutkoordinaterna för de olika paketen vara programmets output.

Figur 1.2 Programmet i den gula rutan är vad projektet syftar till att åstadkomma.

Ett välgjort program i kombination med att robot och transportband är välkoordinerade bör ses som ett mycket hett alternativ för de företag RT Robotics riktar in sig mot.

1.3 Precisering av frågeställningen

Efterforskningen som beskrivs i avsnitt 1.2 förväntas leda till en större förståelse för och klargörande i frågeställningar som:

- Hur kan lösningen (den framräknade lastpallen) representeras visuellt?
- Hur många olika storlekar kan programmet klara av?
- Hur många paketdimensioner behöver programmet vetskap om innan en lyckad beräkning av slutpositioner på lastpallen sker? (En lyckad beräkning sett till hög packningstäthet och att alla paket får plats)

- Kommer lösningen vara bättre och snabbare framräknad än vid manuell packning?

1.4 Avgränsningar

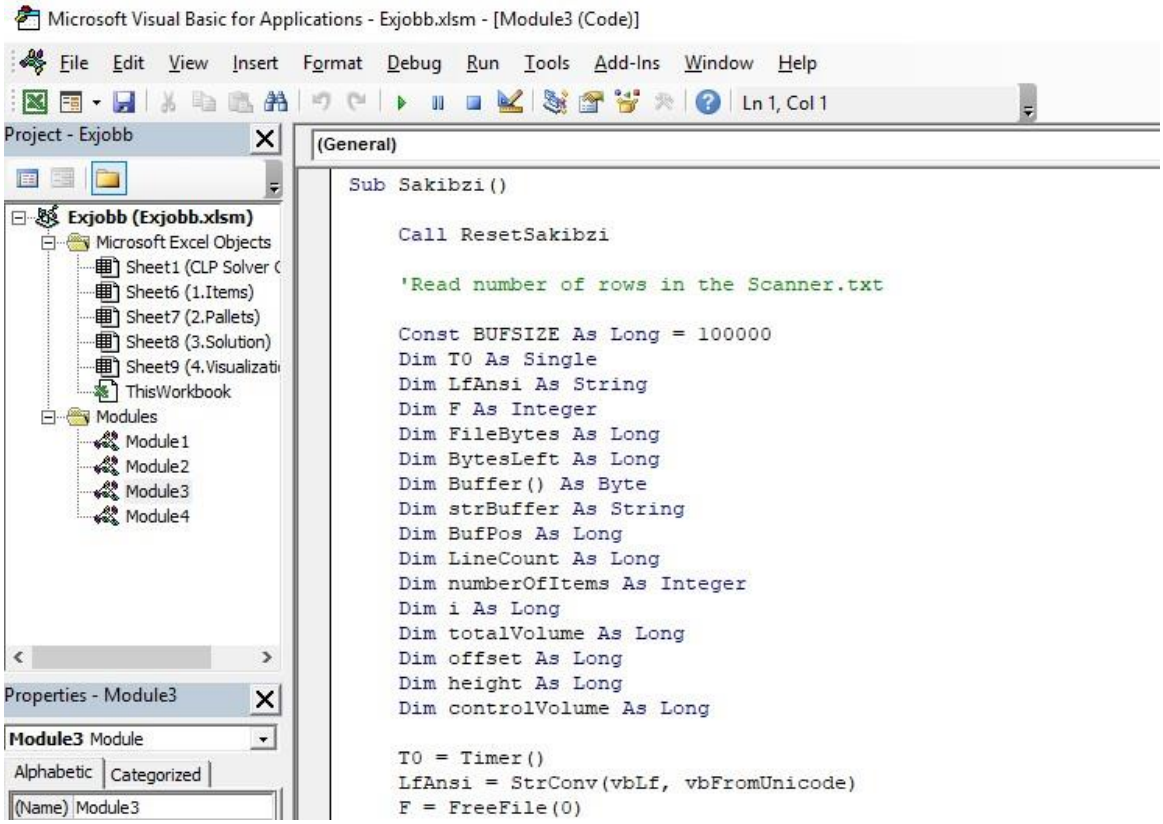
Paketen som behandlas är rätblocksformade, detta för att underlätta programmeringen. Den praktiska delen av denna automatisering lämnas åt företaget. De praktiska delarna är – Hur roboten ska greppa och förflytta paketen (pneumatiska sugproppar, gripdon etc.). Hur paketen ska läsas in (3D-Scanner, kamera etc.), och även förmodad konstruktion samt dimensionering av transportband.

2. TEORI/TEKNISK BAKGRUND

2.1 Visual Basic for Applications - Excel

Rent intuitivt inses styrkan i att Excels utformning. Programmet erbjuder ett brett användningsområde t.ex. för att enkelt kunna visualisera lastpallen med den slutgiltiga lösningen samt visa de inskannade paketen i Excels grafiska interface. Denna möjlighet underlättar för programmerarna när det kommer till att kontrollera att paketen ligger på tillåtna positioner. Att kunna erbjuda en visualisering av den slutgiltiga lastpallen hjälper också projektgruppen att under slutredovisningen av projektet få intressenter (elever, handledare, examinator och företag) att förstå vad programmet uträttar. Detta interface (se figur 2.1) är uppbyggd av rader och kolumner i form av celler. Här kan de behandlade paketen med hjälp av dessa celler tilldelas ett namn (1, 2, 3 osv.) och slutkoordinaterna (x, y, och z) för den beräknade positioneringen på lastpallen enkelt representeras.

Excel erbjuder konventionell programmering i dess egna texteditor. Här kan man på ett liknande sätt som vid programmering med t.ex. C, C#, Java, bygga funktioner som utför iterationer (se figur 2.1). Den stora skillnaden är att den kod och de funktioner som skrivs kan interagera med det ovan beskriva grafiska interfacet. Detta underlättar projektets gång och mycket extraarbete undviks.



Figur 2.1

2.2 Heuristic Algorithms

En heuristisk algoritm är en algoritm som är utformad för att lösa ett problem på ett snabbt sätt, detta genom att offra noggrannhet, precision eller fullständighet för hastighet. Heuristiska algoritmer används ofta för att lösa NP-fullständiga problem, en klass av beslutsproblem. I dessa problem finns det inget känt effektivt sätt att hitta en lösning snabbt och korrekt, även om lösningar kan verifieras när de ges. Heuristik kan producera en lösning individuellt eller användas för att ge en bra baslinje och kompletteras med optimeringsalgoritmer. Heuristiska algoritmer används oftast när approximativa lösningar är tillräckliga och exakta lösningar är nödvändigtvis beräkningsmässigt dyra [5].

En av de vanligaste användningarna av heuristiska algoritmer är att söka och sortera vilket gör detta oerhört centralt för problemet. Till exempel kan denna sökning vara att hämta information lagrad inom en viss datastruktur eller beräknad i sökområdet för en problemdomän, antingen med diskreta eller kontinuerliga värden. När sökningen körs, justerar den sina arbetsparametrar för att optimera hastighet, en viktig egenskap i en sökfunktion. Algoritmen förkastar nuvarande möjligheter om de är sämre än redan funna lösningar. Det tar sökresultat nära målet och följer den nya sökvägen även om det kanske inte fortsätter att leda till det optimala sökresultatet [5]. Detta används då i projektet genom att testa helt olika lösningar för att sedan spara den som anses vara den mest optimala packningen.

2.3 Liknande lösningar

Företagets mål är att automatisera packningen av paket på lastpallar, där den stora skillnaden mot tidigare och liknande lösningar är att de paketgeometrier som behandlas varierar stort. På grund av att RT Robotics tänkta potentiella kunder opererar inom näthandel (läs Zalando, Nelly.com etc.) där de artiklar som ska packas kan vara allt ifrån underkläder till skor behövs således ett nytt program skrivas som klarar av att hantera de högt varierande paketgeometrierna. Om de paket som ska packas är begränsade till ett visst antal olika storlekar löses automatiseringen bäst genom att packa de olika storlekarna var för sig, förslagsvis genom att ha fler än en robot i drift samt tillräckligt antal lastpallar för vardera storlek redo där packningen kan ske.

Då paket av samma storlek ska packas på en lastpall behövs inget avancerat program för jämförelse och uträkning, packningen sker genom att stapla alla paket med samma rotation bredvid varandra. Om alla paketen är rätblocksformade kommer lösningen själv bli ett rätblock där inga tomrum finns. På marknaden finns det gott om automationsföretag som erbjuder automatisering av just dessa fall. Då kommersiella lösningar på det som RT Robotics vill åstadkomma (packning av högt varierande storlekar på samma lastpall) saknas vände de sig till oss i hopp om att få svar på ifall ett program som hanterarar varierande storlekar är genomförbart. Tidigare forskning kring liknande situationer som kännetecknas av stor variation av valmöjligheter (i detta fall paketstorlekar) och lösningar (den packade pallens slututseende), pekar mot ett heuristiskt tillvägagångssätt för att lösa

detta problem [6]. Problemet bör ses som ett optimeringsproblem där det heuristiska tillvägagångssättet möjliggör en tillräckligt bra lösning med mindre kraftansträngning åt beräkning än hos den optimala lösningen (den lösning som ger lägst packningsvolym) [7].

3. METOD

3.1 Inledning av projekt

Under det första mötet med RT Robotics preciserades de frågeställningar som företaget hoppades få inblick (se kapitel 1.3). Projektgruppen ansåg att konstruktion av ett program med ovan nämnda funktion (se kapitel 1.2) var genomförbart. Det beslutades att Projektgruppen efter möte med handledare och efterforskning i lämpligt programspråk skulle kontakta RT Robotics med ett lösningsförslag.

Handledaren underströk vikten av att grundligt efterforska olika programspråk och möjligheter till lösning av projektuppgift. Betydelsen av en god planerad och tydligt upplagd tillvägagångsplan underströk också på detta första möte med handledare. Projektgruppen och handledaren var överens och det bestämdes att tid till brainstorming och därefter en välstrukturerad tidsplan för projektets gång var nästa steg. Handledaren rekommenderade också att läsa på om programmet Wolfram Mathematica då det är ett kraftfullt verktyg, speciellt för visualisering av geometrier.

3.2 Bestämmande av programmeringsspråk

Då projektgruppen under tidigare kurser bekantat sig med C, kändes detta språk naturligt att använda sig av. Det syntax som används i C är bekant och ingen tid skulle behöva läggas åt att förstå ett nytt programmeringsspråk samt dess syntax. Projektgruppen var medvetna av att C i vissa avseenden är ett klumpigt och föråldrat språk [8]. Efter inrådan från handledaren lades också ett par dagar åt att läsa på och fördjupa sig i Wolfram Mathematica.

Efter ett par dagar med fokus på att lösa exempelguider och många timmar av videoinstruktioner i Mathematica, insåg projektgruppen att detta tillvägagångsätt vore allt för omfattande att lära sig. Det finns en del likheter mellan det syntax som används i Matlab men trots det stora skillnader. Att komma upp i en kunskapsnivå tillräcklig för att skriva det program RT Robotics efterfrågade skulle ha tagit för lång tid. Projektgruppen valde därför att använda C för att börja med en prototyp av programmet. Dettas förmedlades till RT Robotics och projektgruppen satte därefter igång med kodningen.

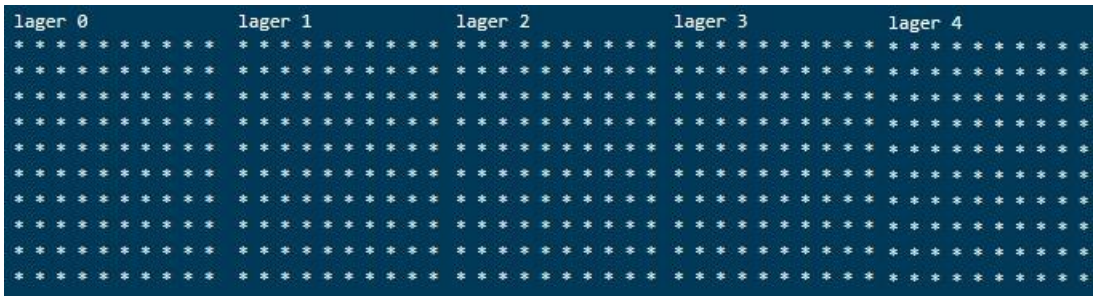
3.3 Prototypprogram i C

Projektgruppen insåg snabbt att det första som behövdes åstadkommas var att beskriva själva lastpallen i C. Principiellt är en lastpall ett stort rätblock med en bottenarea och en höjd som bildar en lastpallsvolym var i packning av paket ska ske. Paketen är rätblocksformade.

3.3.1 Lastpallen realiserad i C

Projektgruppen ansåg att en 3D-array vore lämplig för att beskriva lastpallens geometri. Tanken var att platserna i denna array kunde ha två tillåtna värden. Ett värde som motsvarade att just denna plats var ockuperad av ett paket respektive ett värde för

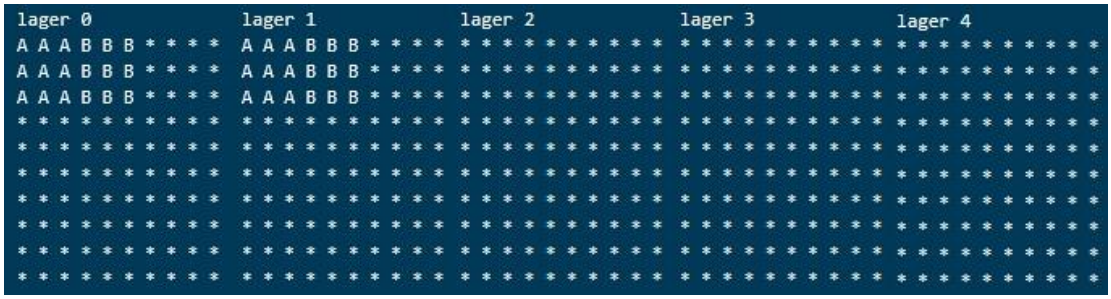
att platsen var tom. Då logiken för en 10 x 10 x 10-matris är den samma som för t.ex. en 10^5 x 10^5 x 10^5-matris kunde denna prototyplastpall beskrivas och hanteras enkelt, för att senare i projektet skalas upp. Alla mått är i längdenheter som senare kan relateras till verkliga fysiska mått, logiken är den samma. I inledningen var alltså lastpallen en array med 10x10x10 platser som kunde anta två värden för att indikera ifall packning var tillåten. Arrayen är från start fylld med tecknet '*' på alla platser vilket betyder att den är helt tom från paket (se figur 3.2).



Figur 3.2

3.3.2 Paketen realiserade i C

Paketen som ska packas är precis som lastpallen själv också rätblock. Låt säga att det första paketet som ska packas har en storlek på 2 x 2 x 2, och indikeras med bokstaven 'A' (första paketet som packas). Idén var då att lastpallsarrayen som från början bara innehåller tecknen '*', på det ställe paketet tilldelas ersätts med bokstaven 'A'. Det andra paketet med notation bokstav 'B', gör det samma på en annan lämplig plats osv (se figur 3.3).



Figur 3.3

3.3.3 Inspiration till konstruktion av program

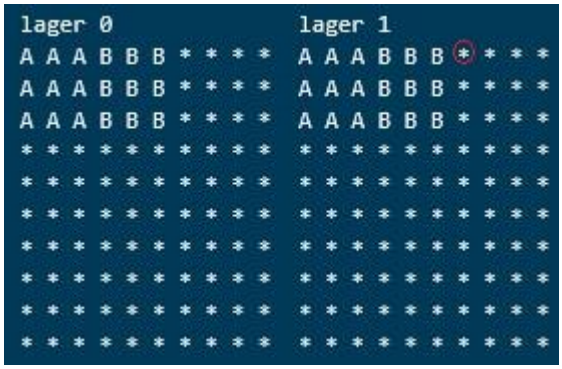
Projektgruppen insåg tidigt under projektets gång att ett spel som tetris, som även finns i 3Dutgåva, har många logiska likheter sett till de objekt som hanteras. Att finna en öppen 3Dtetris-källkod ansågs då kunna vara användbart. Ett sådant program återfanns på internet skrivet i Java. Denna kod bekräftade mest det som redan hade insetts. Att världen liksom lastpallen är en array och att man kan jämföra varje plats i arrayen för att se ifall det ligger något där. Blickarna riktades därefter till manuell packning av lastpallar. Då mänskligheten i över hundra år packat lastpallar finns kunskap och

riktlinjer som förfinats för att effektivisera packningsprocessen. De mest grundläggande begreppen är [9]:

Placera det största paket i ett hörn.
Sträva efter att placera paket med samma höjd jämte varandra, detta för att skapa ett lager ovan med plan yta.
Tunga paket ska packas längre ner på lastpallen för stabilitet.
Undvik överhäng, att paketeten ligger utanför lastpallens avgränsningar.

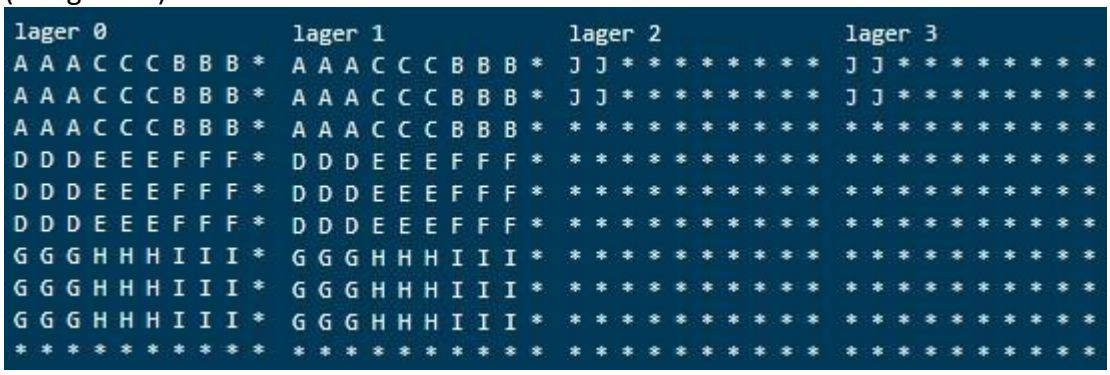
3.3.4 Packningslogik i C

Att packa bottenlagret av lastpallen realiserades relativt smidigt då ingen hänsyn till understöd behövdes göras, bottenlagret är ju som bekant plant. Det första paketet placerades i hörnet och de platser paketet upptog ändrade arrayen på ovan nämnda sätt (med rätt siffra för rätt paket). För att hålla koll på vart tidigare packning har skett och således vart framtida packning kan ske introducerades en pekare som innehåller information om X, Y, och Zkoordinat. Pekaren uppdateras kontinuerligt efter varje genomförd placering av paket (se figur 3.4).



Figur 3.4

Botten av denna virtuella lastpall packades med hjälp av pekaren utan svårigheter (se figur 3.5).



Figur 3.5

Om ett paket är för stort för att få plats, hoppar pekaren upp (Z-koordinaten inkrementeras) och pekar på nästa lager. Proceduren upprepas till dess en fri Z-koordinat återfinns, om så är fallet kontrolleras även ifall plats i X- och Y-led är möjlig. På detta sätt packas lastpallen.

3.3.5 Problematik med C

Problemet med detta tillvägagångssätt var att den kod som krävdes var väldigt omfattande för detta prototypprogram. Även blev det svårt att kontrollera lösningen, då de placerade paketen blev många. Den grafiska representationen av lastpallen blev inte heller optimal.

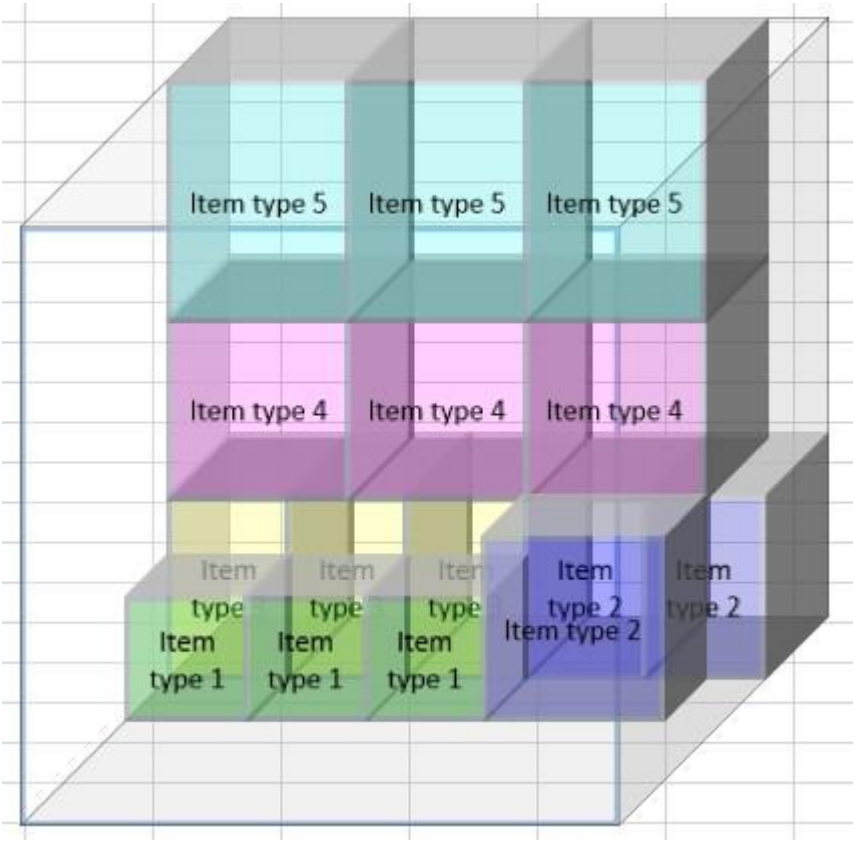
Tanken var ju att skala upp prototypprogrammet till att en plats i arrayen motsvarar en millimeter på en verklig lastpall. Standardmåtten på en EU-pall innebär då att arrayen skulle behövt vara 800x1200xhöjd som beror på hur högt man vill packa. Som följd av den krassa grafiska representationen och faktumet att programmet skulle behöva bli smartare (→ C är ett lågnivåspråk → mycket mer omfattande kodning för grafik och logik) valde projektgruppen att leta efter en annan utvecklingsmiljö.

4. KONSTRUKTION/SYSTEMKONSTRUKTION

4.1 Bakgrund till valt programmeringsspråk samt miljö (VBA)

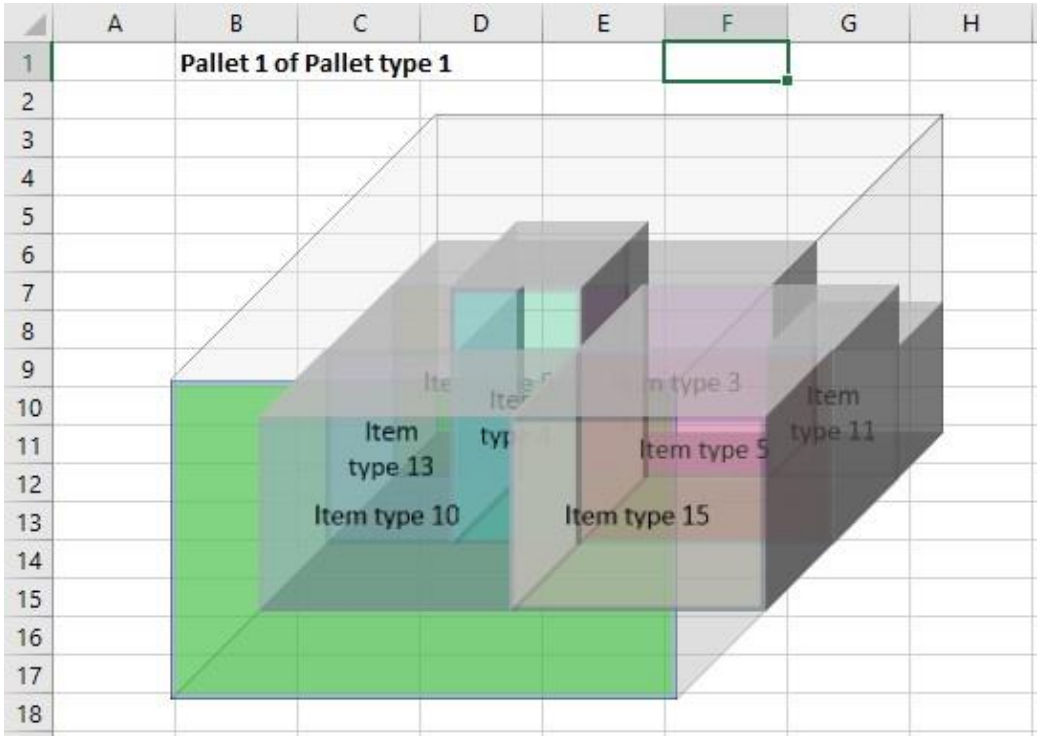
För att förenkla de problem som nämndes i 3.3.5 såg projektgruppen möjligheter som Excel kunde erbjuda. Excel är ett verktyg som med hjälp av sitt kraftfulla grafiska interface, dess programmeringsmöjligheter i Visual Basics for Applications (VBA) samt syntaxlikhet med C erbjöd en alternativ möjlighet till lösning av projektuppgift [10]. På grund av fördelarna insåg projektgruppen att ett program skrivet i VBA, med rådande tidsplan, rimligtvis borde överträffa ett motsvarande i C. Tid lades därför åt att förstå VBA, översätta prototypprogrammet skrivet i C till VBA samt vidareutveckling av programmet i VBA.

Vid efterforskning på internet efter liknande lösningar fann vi ett program anpassat för packning av containrar, framförallt i hamnsektorn [11]. Mycket av den logik vi kom att använda oss av, utgår alltså från detta program. Det finns principiella skillnader i hur man packar en container kontra hur packningsprocessen ser ut på en pall. Vid fallet container handlar det om att bygga lager som fyller väggen inifrån till ut. Detta för att en container har en öppning på en sida (se figur 4.6).



Figur 4.6 Packning av containrar sker genom att man försöker stapla paketen i höjdled först. På bilden har ett lager packats i höjdled och packning av ett nytt lager påbörjats framför det tidigare.

Vid packning av lastpall är det alltså bottenlagret som först ska täckas för att uppnå stabilitet. En tänkt robotarm kan nå lastpallen från alla sidor (se figur 4.7).



Figur 4.7 Här prioriteras utfyllandet av bottenlagret, detta för att packning tillåts från alla sidor på grund av att de är öppna. Tänkt robotarm kan nå hela lastpallen.

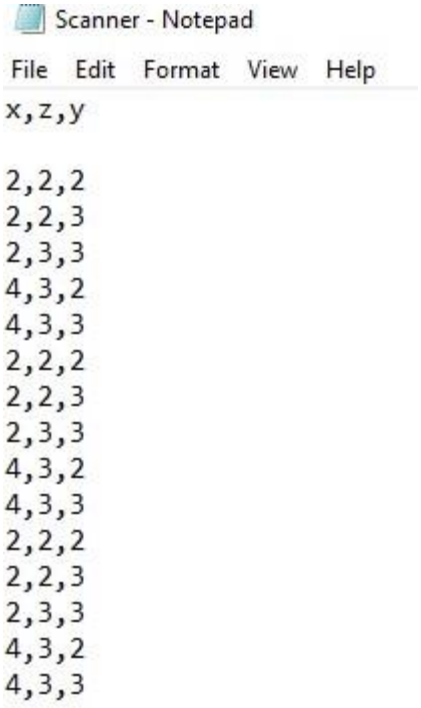
Programmet vi fann, som är licenserat open-source där uppmaning till vidareutveckling ges, erbjuder även en grafisk representation av containern i Excel. Denna funktion liksom logiken vid packningen inkorporerades och modifierades för att uppfylla de önskemål RT Robotics efterfrågade.

Nedan följer arbetsgången för konstruktion av framtaget program.

4.2 Beskrivning av framtaget program

4.2.1 Initiering av paket

För att efterlikna hur en automatisering av packningsprocessen kan komma att se ut användes ett textdokument där paket oberoende av storlek, skrevs in med (se figur 4.8). Informationen som gavs i detta textdokument var paketens bredd, höjd och längd. Vid fysisk inskanning av paket skickas dimensions-informationen kontinuerligt till programmet via en 3D-scanner eller kamera. På grund av detta är följande textdokument en bra representation av den tänkta verkliga automatiserade packningsprocessen.



Figur 4.8

Ju mer information programmet får över de paket som slutligen ska utgöra den packade lastpallen, desto bättre. Därför möjliggörs all den information som textdokumentet innehåller för programmet innan beslut om var paketen placeras på lastpallen. Detta gör då att programmet har fler möjliga lösningar att räkna ut då alla paket redan är inskannade.

4.2.2 Initiering av pall

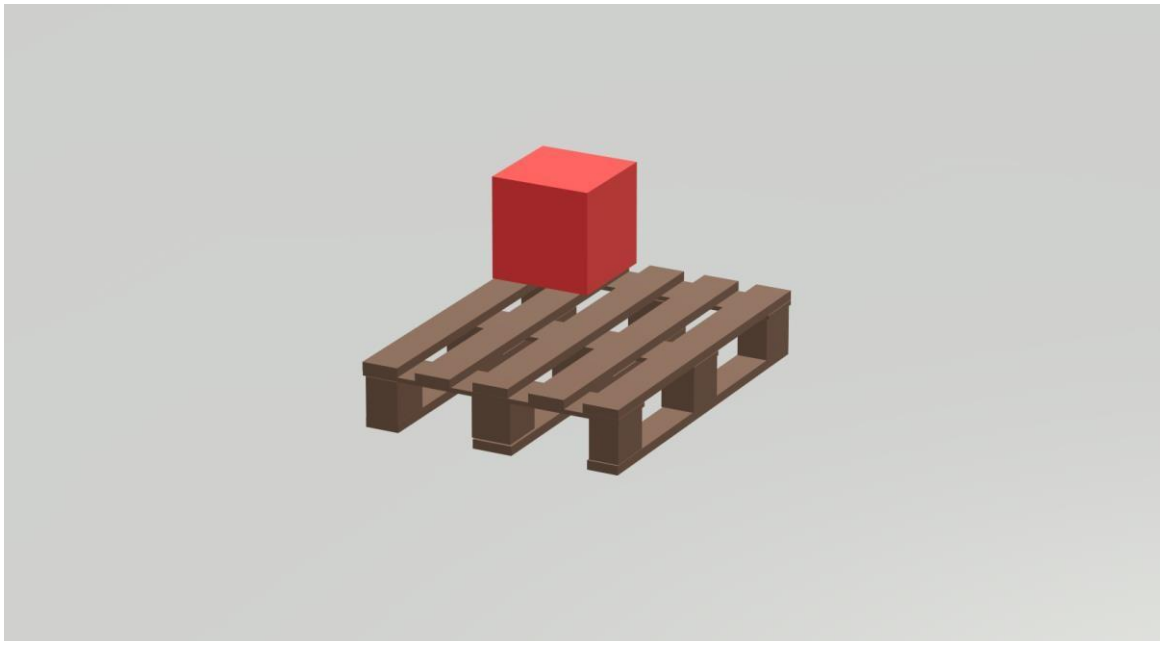
Då paketen är inskannade så ska initieringen av pallen ske. I programmet ges möjligheten att ha valfria mått, men i detta syfte skall som nämnts ovan en EU-pall användas. Denna pall har följande standardiserade mått: Längd: 800mm, bredd: 1200mm. Höjden på pallen kommer att vara beroende på hur högt användaren väljer att packa. I detta fall kommer val av antal paket vara den dimensionerade faktorn för höjden. Denna valmöjlighet kommer ligga hos användaren som själv kan bestämma packningshöjd. Programmet använder sig av en kontrollvolym som alltid håller koll på hur stor volym som paketen utgör. Eftersom denna kontrollvolym är ett teoretiskt mått där packningen är optimal (100 procentig packningstäthet), anges en säkerhetsfaktor. Syftet med denna är att tillåta en packningstäthet som bättre motsvarar en verklig realiserad lösning, som i praktiken aldrig kan vara hundra procent. Varför en packningstäthet på 100% är orealistisk, beror på att en pall med paket vars mått och dimensioner varierar aldrig kan packas utan att tomrum uppstår. Denna säkerhetsfaktor valdes till 20% större än den totala volymen av de inskannade paketen. Detta medför att programmet får ett spelrum att testa olika lösningar som förklaras i efterföljande avsnitt.

4.2.3 Hur packningen sker

Styrkan i den heuristiska algoritmen är att flera lösningar testas innan den optimala lösningen nås. För att få en uppfattning över hur många iterationer placeringslogiken utför kan följande normalfall påvisas. Vid tio paket med en tillåten beräkningstid på tio sekunder så testas ungefär 30 000 olika lösningar innan den mest optimala sparas. Dessa antal lösningar begränsas då av processorhastighet hos den använda processorn.

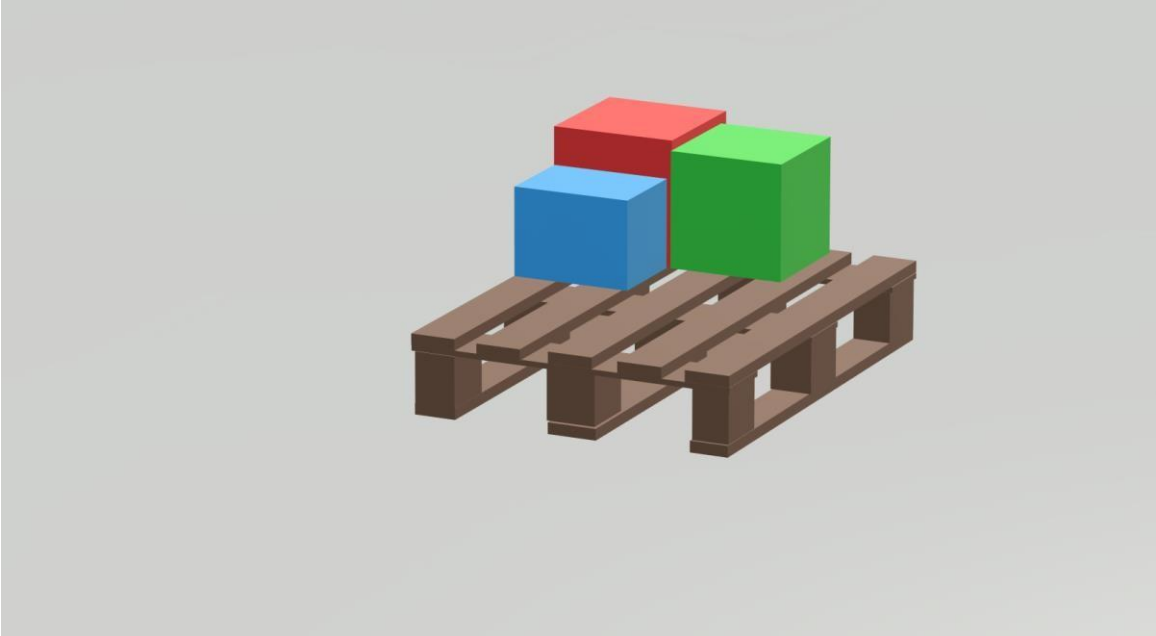
4.2.3.1 Första körningen

Programmet använder en metod där paketen sorteras i storleksordning med avseende på volym under den första körningen av packningsalgoritmen. Det som händer är att det första paketet placeras på pallen, i vänster hörn (se figur 4.9). Det största paketet, med avseende på volym, väljs först.



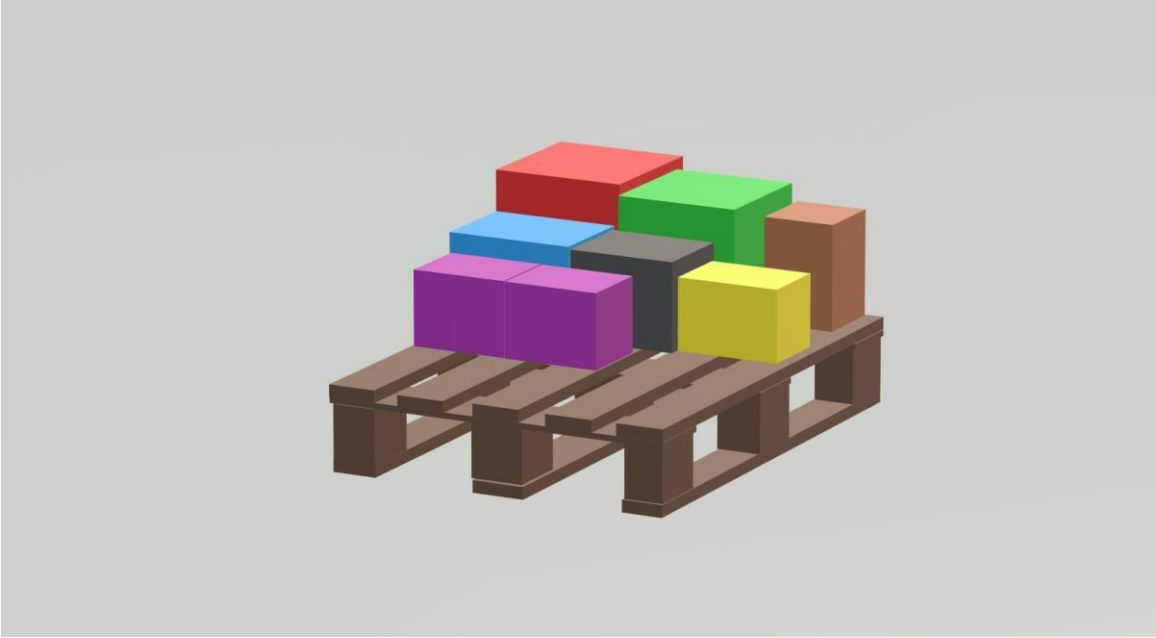
Figur 4.9

Sedan fylls pallen ut från vänster till höger i x-led (bredd) med paket i fallande storleksordning. Detta sker tills nästkommande paket överskrider pallens bredd (se figur 4.10).



Figur 4.10

Då det inte finns plats i x-led söks en ny position fram i z-led (längd) för att sedan fortsätta fylla på i nästa x-led (se figur 4.11).



Figur 4.11

Då bottenlagret är fyllt (till den grad paketgeometrierna tillåter) packas nästa paket där det får plats i y-led (höjd) för att sedan upprepa den ovan beskriva proceduren tills de paket som får plats på pallen är packade i storleksordning. Då ytan inte är jämn görs alltid en kontroll på om det alltid finns ett vertikalt stöd för det paket som skall packas härnäst. Skulle det då för tillfället inte finnas något vertikalt stöd för ett visst paket placeras denna i en lista där det alltid hålls koll på vilka paket som är packade eller

inte. Denna lösning kommer i normalfall (då paketgeometrierna varierar och antalet paket inte är väldigt få) inte vara en optimerad lösning vilket gör att programmet går vidare till nästa steg. Vid resterande körningar är dock lösningen som nåtts under första körningen den bästa lösningen om ej en bättre hittas.

4.2.3.2 Resterande körningar

Vid efterföljande körningar plockas randomiserade paket från de paket som ligger på pallen efter första körningen och adderas till de paket som inte fick plats under den första lösningen. Man kan se denna samling av paket som en temporär lista.

Det som sker sedan är att det tar bort randomiserade paket från pallen och det väljs randomiserade paket från den temporära listan och placeras på pallen från den bästa lösningen (som nu är den första). Då paketen återigen placeras på pallen kan programmet även rotera på paketen. Vad som styr denna paketrotation är också en randomiserad funktion med möjlighet att testa de tre roteringar ett paket (rätblock) tillåter. Detta upprepas tills då inga fler paket får plats på lastpallen. Då borttagningen av de randomiserade paketen sker så görs alltid en kontroll på om det finns något paket ovan eller inte. Detta görs för att inte skapa ”lufthål” i packningen vilket skulle göra att den rasar.

Därefter kontrolleras den packade volymen på pallen och jämförs mot den totala packningsvolymen från den bästa lösningen. Är denna nya packningsvolym lägre så medför detta att denna lösning är bättre och ersätter den förra, annars förblir den tidigare bästa lösningen. Efter detta körs den heuristiska algoritmen om och om igen under den inställda körningstiden. På grund av att sorteringsalgoritmen körs så många gånger kommer en slutgiltig lösning med hög packningsvolym nås.

Från den slutgiltiga bästa lösningen sparas slutkoordinater med respektive ID på paketen till ett nytt textdokument som ska då skickas vidare till den tänkta robot som skall placera dessa paket på den verkliga pallen. Roboten får i och med detta information om vilket paket som ska greppas (packningsordning) och vart paketet ska placeras (slutposition).

4.2.4 Visualisering av virtuell pall

Vid kontroll av de slutkoordinater som fås av den slutgiltiga bästa lösningen krävdes då en visualisering av pallen och de paket som skall packas. Detta gjordes för att enkelt kunna kontrollera lösningen utan att behöva koppla upp hela systemet i praktiken. Denna visualisering sker genom Excel där man enkelt kan använda dess grafiska interface.

5. RESULTAT

Det program som skapades hade den funktion som postulerades under avsnitt 1.2. Syftet var alltså att skapa ett program där input var paketens dimensioner. Beräkning av paketens slutkoordinater på lastpallen, samt ordningen hur dessa paketen ska packas, sker därefter och skickas som output. Detta genomfördes.

Då frågan hur programmet i framtiden ska kommunicera med input och output lämnades åt RT Robotics, användes två textdokument för att representera denna situation. Det ena textdokumentet döptes till Scanner där paketens dimensioner ingick.
Det andra döptes till Koordinater och innehöll information om slutpositionering samt packningsordning på lastpallen.

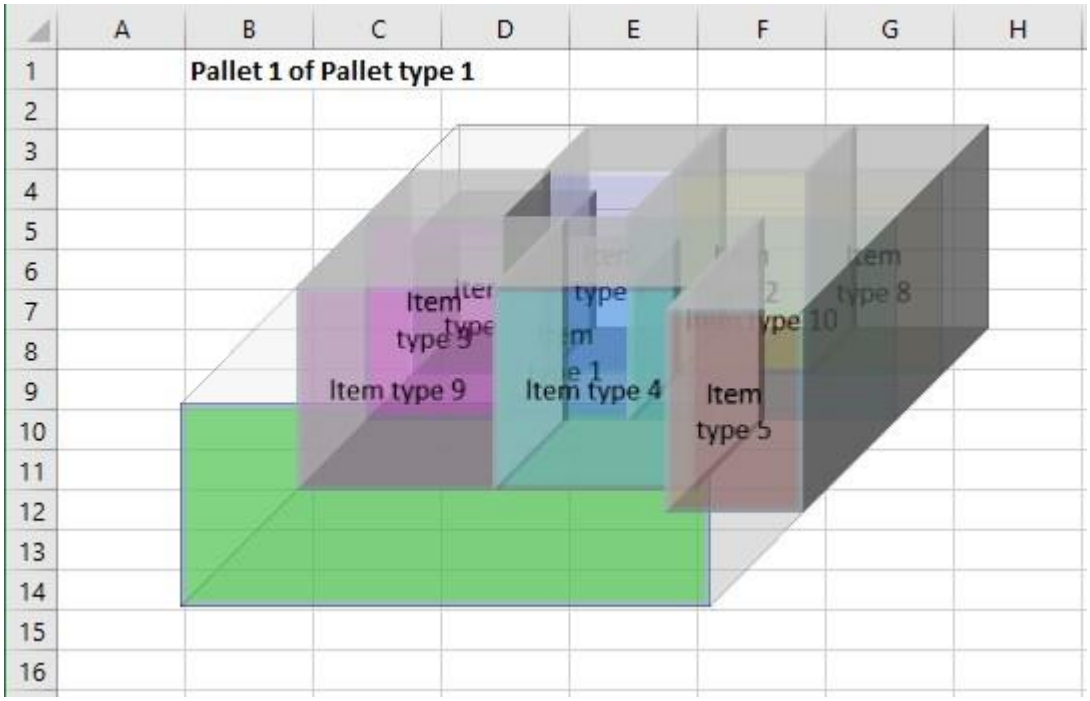
Om en återkoppling till avsnitt 1.3 görs kan de frågor som företaget där ställde besvaras. Nedan listas svaren i fallande ordning:

Hur kan lösningen (den framräknade lastpallen) representeras visuellt?

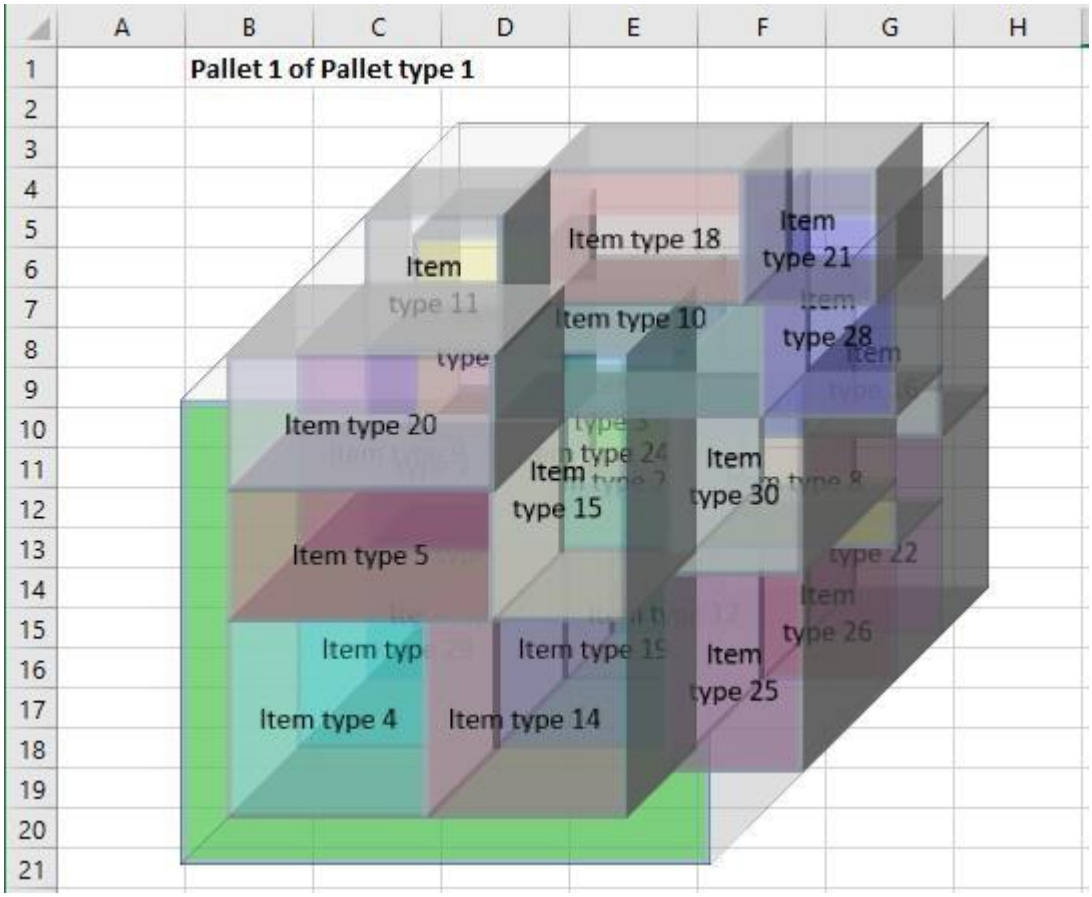
Då ett program anpassat för packningshantering av containrar hittades, där även en grafisk representation av container ingick, anpassades detta för att representera lastpallen. Den visuella representationen var således lyckad.

Hur många olika storlekar kan programmet klara av?

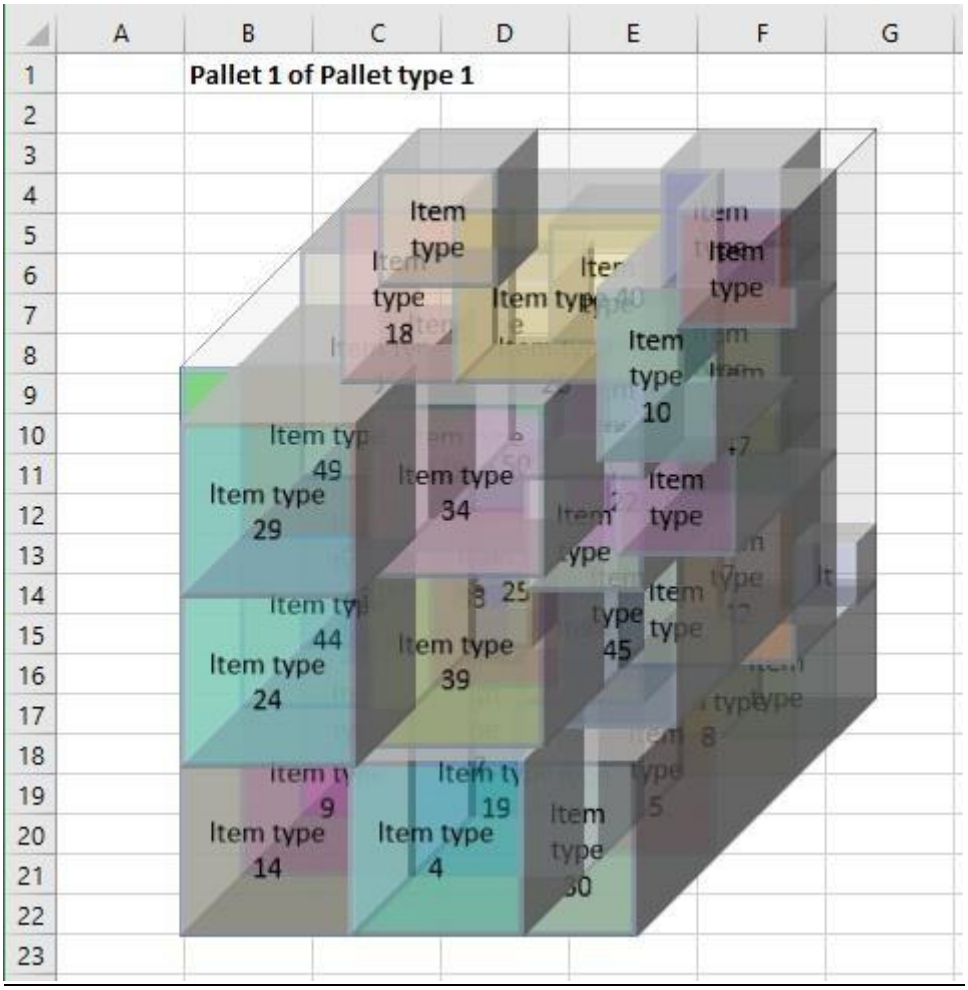
Styrkan i det framtagna programmet är att väldigt många olika storlekar kan hanteras. Följande bilder visar lösningar för 10, 30, och 50 paket (se figur 5.14, 5.15 och 5.16).



Figur 5.14



Figur 5.15



Figur 5.16

Hur många paket måste läsas in innan placering på lastpallen?

Programmet konstruerades med fokus på hantering av stor variation av paketgeometrier. Därför krävs det att alla paket behöver vara inlästa innan uträkning sker. Detta för att nå en så optimal lösning som möjligt.

Kommer lösningen vara bättre och snabbare framräknad än vid manuell packning?

Rekommendationen är att programmet ges en tid på en sekund per paket. Detta på grund av att programmet på denna tid tillåts testa flera olika lösningar för att nå en optimal slutgiltig lösning. 100 paket tar alltså 100 sekunder att räkna ut. Då paketen är flera till antalet och dimensionerna varierande märks fördelen med programmet. Efter programkörningen finns en lösning som har en hög packningstäthet. Eftersom programmet ges en tid på en sekund per paket för att beräkna slutpositionering, så är det osannolikt att mänsklig packning under samma tidsram kommer att överträffa den hos programmet. Den praktiska utformningen av packningen (robot, inskanningsteknik, transportband) kommer att avgöra om den automatiserade packningen i slutändan blir snabbare än motsvarande mänsklig.

Om roboten med hjälp av denna lösning packar effektivt och transport av paket till robot konstrueras effektivt, kommer systemet som helhet överträffa manuell packning. Tid behövs inte läggas åt att under packningen justera och packa om paket.

6. SLUTSATS/ANALYS/DISKUSSION

Det framtagna programmet är väldigt användbart då de paket som ska packas är många och storlekarna stort varierande. Den heuristiska metoden uträkningen sker med har ingen övre gräns på antal paket. Den upprepar uträkningen under den tid som programmet blivit tilldelad. Detta är styrkan i programmet.

Frågan man dock kan ställa sig är hur väl detta program är anpassat mot den verkliga situationen i lagerlokaler. Är det rimligt att ha möjligheten att scanna in alla paket innan packning sker? Detta medför att långa och komplicerade transportband behöver utfylla lagerytan. Ofta är denna lageryta begränsad och således ett system med detta program otänkbart, det finns helt enkelt inte plats. Att då istället konstruera ett program som inte behöver information om alla paket innan beslut om slutpositioner beräknas kan vara mer användbart. Packningstätheten för en lastpall packad med ett sådant program kommer inte i normalfall vara lika hög, då information om alla paket saknas. Frågan är dock om packningstätheten kommer vara tillräckligt hög för att kunna konkurrera med den packningstäthet en manuell packning levererar. Om beslut tas efter t.ex. fem inskannade paket kan även den yta som upptas av inläsningsprocessen (transportband, scanner och dylikt) göras mindre och då mer lättapplicerad ute på verkliga lagerlokaler, där utrymmet ofta är en begränsande faktor.

På grund av tidsbrist hos både RT Robotics och projektgrupp kunde inte heller ett testsystem (scanner, programkod och robot) realiseras. En ytterligare bidragande faktor till att detta uteblev är även ovan nämnda programbegränsning. RT Robotics kände att de inte finns utrymme för inskanning av alla paket innan placering på lastpallen, hos de potentiella kunder som var påtänkta. För att investera tid och pengar i realisering av ett testsystem önskade RT Robotics att packning kunde ske efter t.ex. fem paket. Ett sådant testsystem skulle kunna visas upp på mässor för att övertyga potentiella kunder.

RT Robotics ville använda sig av en kamera för inskanning av paket. Denna kamera hade fördelen att den var billig men nackdelen att den endast kunde hantera ett paket åt gången. I början av projektet fanns det även tankar om att använda sig av en mer sofistikerad inskanningsteknik (3D-Scanner) som kunde hantera fler paket åt gången. Nackdelen är dock att den ofta är tio gånger så dyr. Fördelen med att fler paket kan skannas in åt gången är att alla paket inte behöver ligga uppradade på ett långt transportband. Då behöver inte transportbanden uppta en så stor del av lagerutrymmet och programmet i detta projekt kunde i så fall användas.

Verkligheten är dock att hela systemet inte får kosta för mycket, vilket användandet av en dyr och sofistikerad 3D-Scanner medför. Manuell packning kan då trots sina ekonomiska brister fortfarande vara mer fördelaktig och beslutet om att investera i en automatisering mindre lockande hos de kunder RT Robotics riktar sig mot.

7. FÖRSLAG TILL FRAMTIDA ARBETE

Det projektgruppen rekommenderar ett framtida arbete att undersöka är möjligheten till att konstruera ett program där alla paket inte behöver läsas in innan uträkning av slutkoordinater sker.

Vad skulle packningstätheten bli om ett sådant program konstruerades med ovanstående programlogik (heuristic)?

Finns det alternativa sätt att lösa packningslogiken på som inte är lika beroende av information om alla paket innan uträkning sker?

En intressant tanke är också huruvida ett framtida liknande program kan inkorporera artificiell intelligens. Kan programmet bli ”smartare” under programkörning och ”lära” sig se mönster i den följd paketen frammatas på transportband? Med hjälp av sannolikhetslära och mönsterigenkänning kan denna artificiella intelligens medföra till en snabbare och bättre packning?

Ett framtida arbete kan även undersöka de praktiska begränsningar som uppstår vid automatisering av packningsprocesser. Kan inskanningstekniken göras effektivare? Behöver alla paket ligga på rad på ett transportband för att inskanning kan ske? Det är just detta faktum som gör programmet i detta projekt ineffektivt då långa transportband och medföljande utrymme uppkommer.

8. REFERENSER

[1] Lastpallen som förändrade historien, Dagens arbete 2010, https://web.archive.org/web/20140903092203/http://www.da.se/home/da/home.NSF/files/DA NOV13_17.pdf/%24FILE/DANOV13_17.pdf (Acc 2019-05-01)

[2] The comprehensive guide to freight shipping services, Mach 1 Global Services Inc 2018,https://www.mach1global.com/what-are-freight-shipping-services/ (Acc 2019-05-02)

[3] Robotar har många fördelar, Danica Kragic Jensfelt 2015, https://www.di.se/di/artiklar/2015/7/22/robotar-har-manga-fordelar (Acc 2019-05-02)

[4] Här är hetaste robotarna som slåss om kunderna, Marie Palm NyTeknik 2015, https://www.nyteknik.se/automation/har-ar-hetaste-robotarna-som-slass-om-kunderna6393125 (Acc 2019-05-10)

[5] Heuristic Algorithms, ChE 345 2014, https://optimization.mccormick.northwestern.edu/index.php/Heuristic_algorithms (Acc 201903-22)

[6] Automated packing systems: Review of industrial implementations, Paul F. Whelan, Bruce G. Batchelor, http://www.cipa.dcu.ie/papers/SPIE93.pdf?fbclid=IwAR1oUhh_xgF_S7HHFTpSSr5p5DS0g4ClKapkf0Ad8nYvlzcuRnMzhkoo4QU (Acc 2019-06-28)

[7] Handbook of Metaheuristic, Michel Gendreau Polytechnique 2010, https://orbit.dtu.dk/files/5293785/Pisinger.pdf?fbclid=IwAR3e4LdUm7JxTq5Qv4y3CkqFtAXBstWJn_QNNHNwyqQx8lRo59O3QZuksMk (Acc 2019-06-28)

[8] Why C and C++ are Awful Programming Languages, Radford 2017, https://www.radford.edu/ibarland/Manifestoes/whyC++isBad.shtml (Acc 2019-04-12)

[9] Packing Guide, DHL 2019, https://www.dhl.com/content/dam/downloads/g0/express/shipping/packaging/dhl_express_large_palletised_packing_guide_en.pdf (Acc 2019-04-12)

[10] Excel VBA reference, Microsoft 2018, https://docs.microsoft.com/enus/office/vba/api/overview/excel (Acc 2019-04-12)

[11] CPL Spreadsheet Solver, Güneş Erdoğan 2019, http://people.bath.ac.uk/ge277/index.php/clp-spreadsheet-solver (Acc 2019-04-12)