

Optimering av sekvenser i produktionsplanering

Erik Kronberg
IEA, Lunds tekniska högskola
Mars 2021

De flesta människor arbetar utefter något slags schema vare sig det är i skolan eller på arbetet och har förmodligen någon gång ställt sig frågan "hur svårt kan det vara att göra ett schema". Schemaläggning kan bli oerhört komplext när många olika scheman ska integreras med varandra och passa ihop på ett så bra sätt som möjligt. På samma sätt är det inom tillverkningsindustrin - all produktion ska planeras och schemaläggas och givetvis eftersträvas en så effektiv planering som möjligt. Med hjälp av smarta optimeringsalgoritmer kan ett sådant schema erhållas.

ALL produktion följer någon slags planering vare sig det är en kontinuerlig eller diskret process, en stor ståltillverkare eller ett litet snickeri. Många gånger görs planeringen manuellt av en planerare - ett arbete som kan vara komplicerat eftersom det ofta finns många aspekter att ta hänsyn till. Leveransdatum som ska hållas, material som ska finnas tillgängliga och ställtider som uppstår eller andra regler som måste följas. Detta jobb kan emellertid göras av en dator med en smart optimeringsalgoritm - vilket är precis vad som har gjorts i det här arbetet.

Syftet med arbetet var att implementera och jämföra två stycken optimeringsalgoritmer för att optimera produktionsscheman. Genom att optimera ett schema kan den totala ställtiden som schemat ger upphov till minimeras, samtidigt som leveransdatum och materialtillgång tas med i en beräkning. Ställtid uppstår typiskt vid produktbyte och kan utgöras av till exempel färgbyten i färgsprutor. Genom att minimera dessa kan företaget i fråga nå en mer effektiv produktion och således spara in på både miljö och resurser. Produktionsscheman illustreras ofta som gantt-scheman, se exempel i figur 1.

Detta arbete fokuserar på en speciell uppsättning av maskiner i industrin som kallas parallella resurser - vilket är en vanlig konfiguration som innebär att det finns ett flertal likartade resurser som kan utföra samma arbete. Till exempel kan det vara ett bryggeri som har tre fyllningslinjer, eller en biltillverkare som har tio lackeringsrum. Genom att ha ett flertal resurser att välja mellan ökar frihetsgraden på planeringen och även komplexiteten.

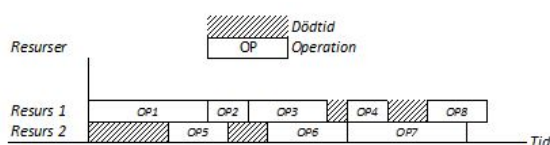


Fig. 1: Gantt-schema

Alla optimeringsproblem som handlar om sekvenser i någon mening, kan modelleras som ett handelsresandeproblem. Så kan även detta - mer specifikt som ett handelsresandeproblem med flera säljare. Låt städerna bytas ut mot operationer, säljarna mot resurser och distansen som vanligtvis ska minimeras mot en kostnad - nämligen den kostnad som uppstår till följd av eventuell ställtid, leveransförse-ningar eller annat. Genom denna modellering kan välkända lösare såsom genetiska algoritmer och simulerad kylning användas för att hitta ett (nästan) optimalt schema.

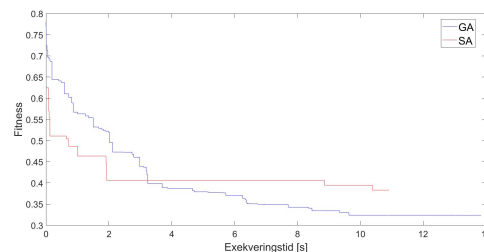


Fig. 2: Resultat från en optimering med 30 operationer och 3 resurser

Handelsresandeproblem är komplexa och ingen universallösning har hittats till dessa men med hjälp av heuristiska optimeringsmetoder såsom de två nämnda kan en åtminstone nära-optimal lösning hittas på kort tid.

För det här fallet av schemalägningsproblem användes just en genetisk algoritm och simulerad kylning. Båda algoritmerna har en strategi som är hämtad från naturen; den genetiska algoritmen använder sig av en population med individer (lösningar) och genom artificiella avkommor letar den sig fram till bättre sådana genom att mixtra med generna och simulerad kylning ska efterlikna en glödningsprocess i till exempel metall.

Genom att testa algoritmerna och jämföra dessa med olika inställningar av deras parametrar erhöles ett resultat som påvisade att simulerad kylning var snabbare men den genetiska algoritmen hittade bättre lösningar. Vid upprepade exekveringar visade sig den genetiska algoritmen även vara mer stabil. I resultatet presenteras också hur antalet operationer och resurser påverkar exekveringstiderna för algoritmerna. Figur 2 visar ett typiskt resultat från en simulering, där simulerad kylning (SA) snabbt hittar bättre lösningar men också konvergerar tidigt medan den genetiska algoritmen (GA) fortsätter att hitta bättre lösningar långt efter.