orbit # 35

November • 2020

Aarhus OR Days 2020 • Mathematical Optimization for facility location under COVID-pandemic • Splitting Social Networks to Limit the Spread of COVID-19 • Quayside Planning in Container Terminals (...)

ORbit

medlemsblad for
Dansk Selskab
for Operationsanalyse
og
Svenska Operationsanalys-
föreningen

Redaktion:
Ansv. Julia Pahl (jp)

Dansk Selskab for Operationsanalyse
(DORS)
c/o DTU Management
DTU Management, bygn. 358
Danmarks Tekniske Universitet
DK-2800 Kgs. Lyngby

E-mail: dors@dorsnet.dk

Næste deadline:
10. March 2021

Tryk:
Ecograf Gruppen

Oplag: 375
ISSN 1601-8893

Forside: Billede leveret af colorbox.
com.

Editor

Dear Reader,

The Covid-19 Pandemic keeps us in its chocke-hold with another lock-down of work and social life closely around the corner. We are all required to stay at home and practice social distancing. Therefore, I hope that this ORbit edition can cheer you up a bit with interesting articles on, e. g., how Operations Research can help also little businesses to fulfil distance regulations due to Covid-19 in an optimal way.

The Aarhus OR day 2020 was helt online permitting a lot of people to participate online which was a nice experience even though of course we all would have preferred to meet face-to-face.

In this ORbit ediction, Monica Fischetti shows us how to deal with social distancing regulations due to Covid-19. Niels-Christian Fink Bagger, Evelien van der Hurk, David Pisinger look at the Pandemic from another angle providing decision support for policy makers regarding contact limitation measures.

Getting a little bit away from our ”new reality” of social distancing, articles on how to model diversity of solutions by Linnea Ingmar or the DORS Price winning article of Peter Emil Tybirk on a primer on reinforcement learning for combinatorial optimization should be able to help you. Moreover, we have an article on quayside planning of container terminals provided by Rasmus Riber as well as one that shows us how to win an international timetabling competition given by Dennis S. Holm, Rasmus Ø. Mikkelsen, and Thomas Stidsen. Last but not least, Daniele Gammelli shares with us his machine learning approach for shared mobility demand prediction.

Please pay also attention to the call for the next DORS Price Winning Thesis and next year’s AOO! I wish you all a good reading! Julia Pahl (Editor)

Aktuelt om DORS

God læsning

Medlemsskab
Kontingentsatser

Personlige medlemmer

(incl. ph.d.-studerende):	270 kr./år
Studerende:	0 kr./år
Firmamedlemmer:	3200 kr./år
Institutmedlemmer:	1800 kr./år

Indbetales på Giro 9123865 (reg.nr. 1199)

Julia Pahl, redaktør

Sekretariat
Dansk Selskab for Operationsanalyse
(DORS)
c/o DTU Management
DTU Management, bygn. 358
Danmarks Tekniske Universitet
DK-2800 Kgs. Lyngby

e-mail: secretary@dorsnet.dk
Internet: www.dorsnet.dk

Svenska operationsanalysföreningen

Nu är hösten, och snart till och med vintern, här. Förra gången jag förberedde min text för ORbit var COVID-pandemin i sin startfas, och då hoppades och trodde nog de flesta av oss att livet skulle återgå till sin normala lunk framåt hösten. Att arbete, undervisning och konferenser skulle vara tillbaka på ett någorlunda normalt sätt igen vid det här laget. Under sommaren och början på hösten började vi gradvis förstå att så inte skulle bli fallet, och vid det här laget vågar vi knappt ens hoppas på normalitet under första halvan av 2021.

Många utmaningar kvarstår innan pandemin är över, men samtidigt har vi också lärt oss att leva med färre fysiska och fler virtuella möten i arbete och vardag. Detta skifte har gett oss nya insikter och utmaningar - jag är nog inte den enda som optimistiskt har bokat in sig på flera lätt-tillgängliga virtuella konferenser utan att reservera tillräckligt med tid i kalendern för att faktiskt delta på ett bra sätt, eller varit helt utmattad efter dagar fyllda med virtuella möten.

Arbetet i SOAF har givetvis också påverkats av pandemin. Redan innan pandemin hölls de flesta styrelsemöten via telefon, och faktum är att dessa snarare har blivit mer fysiska nu eftersom vi har börjat använda video för det mesta. Styrelsen har under året arbetat med olika initiativ för att ge våra medlemmar ännu mer valuta för medlemsavgiften, och vi hoppas snart kunna dela mer information med er om detta. Vi har även tittat på hur vi ska ge kontinuitet till vårt uppskattade doktorandnätverk, och haft de första diskussionerna om nästa års konferens och årsmöte.

Mattias Grönkvist, Ordförande, SOAF

Indhold

Aktuellt om SOAF

Individuellt medlemskap: avgiften är 160 kr/år.

Företagsmedlemskap:
Avgiften för företaget/institutionen beror på antalet OR-intressenter enligt följande tariff:
Akademiska institutioner: 1600 kr/år
1-5 intressenter: 1600 kr/år
6-10 intressenter: 3200 kr/år
över 10 intressenter: 4800 kr/år

I företagsmedlemsskapet ingår 2-6 exemplar av ORbit, beroende på företagets storlek.

Betala in på postgiro: 19 94 48-2 (Svenska Operationsanalysföreningen)

e-mail: ordforande@soaf.se

Internet: www.soaf.se

By Julia Pahl

Aarhus OR Day 2020

This year's OR Aarhus Day 2020 took place online due to Covid-19 that holds the world under breath and at home since March 2020.

This is the right time to be glad for the technological improvements that took place in recent years with platforms from Zoom to MS Teams, just to mention a few, that allow the community to reach out to even more people as under normal circumstances. In the early minutes after starting off, 40 people had already logged on the event peaking at ca. 60 at some points in time.

The welcome speech to the OR Day 2020 was held by Ata Jalili Marand, the representative of the Cluster for Operations Research, Analytics, and Logistics (CORAL) at Aarhus University, and the welcome speech for DORS and some introductory words on this organization given by Dario Pacino, the president of DORS.

The first talk was given by Mia Bredal, Senior Manager in Product Development at Arla on *Savings Through Supply Chain Optimization and Transformation.* We got a lot of insights into how different toolsets can help reducing complexity of workstreams in various areas of an organization from procurement,

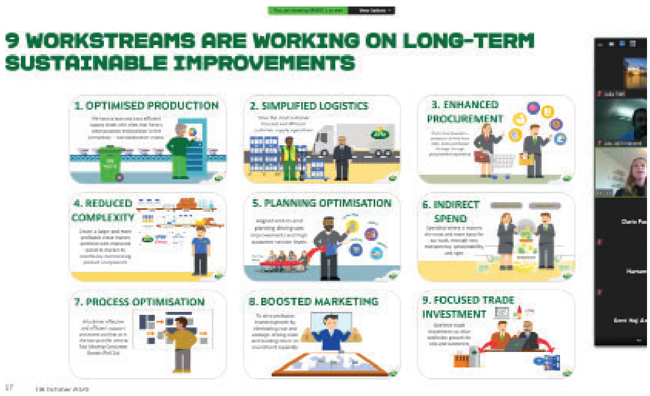

Figure 1: Mia Bredal talking about Sustainable Supply Chain Optimization and Transformation

production, logistics to marketing and trade investments. Besides, Mia showed her excellent talent of motivating people to participate in the talk via Kahoot.

Figure 2: Jacob Roldsgaard Poulsen and Aksel Poulstrup Presenting Sales and Operations Planning performed at Danish Crown

Jacob Roldsgaard Poulsen, Senior Manager, Global Demand Planning gave together with Aksel Poulstrup, Director, Sales and Operations Planning at Danish Crown the second talk of the day on *Business transformation in Danish Crown Foods* by implementing the Sales and Operations Planning Concept. It was interesting to learn how Danish Crown approaches sales and operations planning at such a great company.

After the lunch break, Victor Bloch, Senior Consultant at INVERTO, a BCG Company, introduced the participants to how consulting is lived at the company with the talk on *Consulting: from acquisition to implementation*. Victor presented concrete examples of the company's consultancy projects and gave the audience an insight on how fast, as a consultant, one needs to turn to an expert within a field from knowing nothing about it.

The last talk of the day before the time to zoom-network with the companies and the participants was given by Brian Bruhn Sørensen, Senior Group Process Consultant, SC Process

Excellence at Grundfos on *Analytics in Sales, Inventory and Operations Planning*. Brian explained the supply network planning at Grundfos with a focus on inventory management. His presentation was an excellent example of how operations management tools and techniques are implemented in a world-class company.

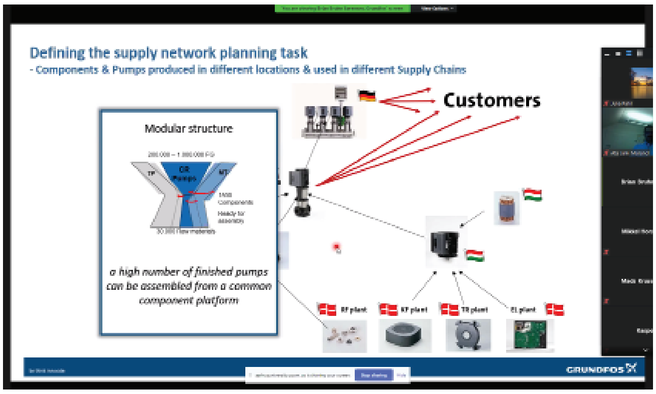

Figure 3: by Brian Bruhn Sørensen talking about Analytics in Sales, Inventory, and Operations Planning at Grundfos

Afterwards, time was invested in zoom networking sessions for INVERTO and Grundfos. Unfortunately, only a few students joined the sessions.

We are looking forward to the next Aarhus OR Days in 2021 crossing our fingers that we will be able to meet, again, in person to discuss and learn about how OR is applied in various companies and share some small talk face-to-face while maybe permitting some online-participation for those who are not able to be there in person.

DORS' firma- og institutmedlemmer

Institutmedlemmer

- Datalogisk Institut, Københavns Universitet
- Institut for Virksomhedsledelse og Økonomi, Syddansk Universitet
- Institut for Matematiske Fag, Aarhus Universitet
- Afdeling for Anvendt Matematik og Statistik, IMF, Københavns Universitet
- CORAL, Aarhus University
- DTU Management, Danmarks Tekniske Universitet
- Department of Materials and Production, Aalborg University

Firmamedlemmer

- A.P. Møller – Mærsk
- Copenhagen Optimization
- DONG Naturgas A/S
- DSB S-tog
- Hermes Traffic Intelligence
- Københavns Lufthavne A/S
- MaCom A/S
- MOSEK
- Novo Nordisk (CMC Clinical Supplies)
- Optivation
- PDC A/S
- PostNord
- QAMPO
- Rapidis
- Transvision A/S
- Trapeze Group Europe A/S

By Martina Fischetti

Mathematical Optimization for facility location under COVID-pandemic

Abstract: Moments of crisis are also opportunities to look at the world with new eyes. This is how we discovered an analogy between the optimization challenge of locating turbines offshore and the one of locating facilities to ensure social distancing and safety during COVID-pandemic. This work shows an example of how Operations Research can help businesses and customers during this challenging period. In particular, we show that using mathematical optimization one can increase profit for businesses (fitting more tables in a restaurant, umbrellas on a beach, etc.) while also increasing safety for customers.

1 Introduction

The spread of viruses such as SARS-CoV-2 brought new challenges to our society, including a stronger focus on safety across all businesses. In particular, many countries have imposed a minimum social distance between people in order to ensure their safety. This brings new challenges to many customer-related businesses (such as restaurants, offices, etc.) on how to locate their facilities under distancing constraints. Can we then use Operations Research to help business-owners to serve more customers while satisfying social distance regulations? Or, in other words, can we find facility layouts that maximize the usage of space within safety regulations? This is a very relevant question for many businesses such as restaurants, pubs, etc. as more facilities (tables, seats etc.) translate in higher profits.

We also went a step forward asking ourselves: among layouts with the same number of facilities that respect minimum distance constraints, are there layout configurations that are safer than others? Indeed, let us look at the toy example of Figure 1. This shows a regular layout for a restaurant: regular layouts are typical of manual solutions, as they represent a simple strategy to locate facilities within the safety distance regulations. All tables, indeed, are located according to social distancing constraints... but does this actually mean that all tables are equivalent in terms of safety? If you would have to sit in such a restaurant, where would you sit? Our previous experience on a different yet similar problem (the wind farm layout problem) suggests that tables in the center are more risky as they are subject to virus spread from all directions. We therefore focus on the case of optimizing facility location to maximize safety. In this second case we have a fixed number of customers that we can locate in the area, and the challenge is to minimize virus spread.An example of this application could be a restaurant with a limited kitchen capacity and a big space available. Here the challenge is not anymore to fit as many tables as possible, but to design the safer table layout. To properly answer the above questions we used our experience in wind farm design, exploiting a parallel between the two situations.

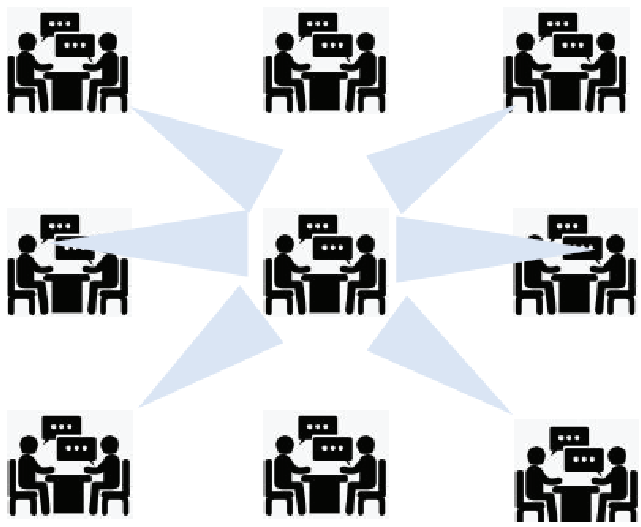

Figure 1: A restaurant has placed all tables on a regular grid with a fixed distance between tables to respect social distancing limitations. Is it equivalent, in terms of safety, to sit at any table?

2 Optimal social distance and optimal wind farm layout: different yet similar

In the last years we have been working on the offshore wind farm layout optimization problem, which consists in deciding where to place turbines in a given area in order to maximize production while reducing costs (refer for example to [1,2]).

A key aspect in the optimization is to take wake effects into account. The wake effect is the interference phenomenon for which, if two turbines are located one close to another, the upwind turbine creates a shadow on the one behind. This is of great importance in the design of the layout since it results in a loss of power production for the turbines downstream, that are also subject to a strong (hence damaging) turbulence. Also, nearby wind farms might need to be considered in the optimization as they can also interfere with the new one.

In practical applications, a minimum and/or maximum number of turbines to be located in the area can be imposed, together with a minimum distance between turbines (to avoid the blades clash, and also for turbulence considerations). There can be obstacles within the offshore area (such as natural reserves, preexisting infrastructures, bad seabed areas etc.), which are areas where the turbines cannot be located. The wind farm layout optimization problem therefore consists in locating a given number of turbines (or as many as profitable) in a given area, ensuring a minimum distance between turbines and minimizing the interference between them.

What about our problem about social distancing in a public place such as a restaurant? It also consists in locating a given number of facilities (tables or customers) in a given area, ensuring a minimum distance between them (legal or recommended social distance) and minimizing the potential virus spread between facilities. Our experience on the wind farm layout problem shows that optimal layouts tend to use the borders of the available area, where the turbines create less interference to other turbines. If we therefore look back at the initial question of Figure 1, the tables on the borders are to be preferred, as they suffer from virus spread coming from fewer directions. At the same time, a smarter location of tables would have placed more tables on the borders, as we will also see in our tests of Section 3. Our results for the wind farm layout problem also show that traditional manual layouts where turbines were placed on a regular grid are highly sub-optimal as significantly higher production can be achieved by a less-regular but smarter (i.e., optimized) placement of turbines. The resolution of the problem in practical applications is far from trivial, and state-of-the-art mathematical optimization techniques have proved to make a huge impact in the practical resolution of the problem, resulting in savings in the order of hundred million Euros [2].

Figure 2: Wake effect on a real wind farm (Horns Rev 1): regular layouts like the one in the picture can be very inefficient for certain specific wind scenarios, resulting into a greatly reduced energy production for the whole wind farm. [Source: Vattenfall]

Figure 3: Are virus spread and wake effect so different after all?

We used our previous work and expertise in wind farm design [1] to write a mathematical formulation of the facility location problem and ad-hoc heuristics to solve it. We also propose a virus spread model (equivalent to an interference function between facilities) and show its impact on the final layout. More details can be found in [3].

3 Applications

We show here a real case in Denmark to test the impact of optimizing table location in the restoration business: the brewpub Brus, in Copenhagen. Brus has an available outdoor area where tables can be placed. Using online satellite views (first plot of Figure 4), we identified the available area for tables (red area in the second plot of Figure 4). Note that the presence of a tree does not allow to place tables in the middle part of the area, which was therefore excluded from the set of available positions. Being able to easily exclude areas from the optimization may also allow for the definition of free corridors for customers or personnel movements.

3.1 Fitting more tables under social distancing rules

The first problem we would like to solve is to place as many tables as possible in the available area, while of course complying with the minimum distance between tables imposed by country's regulations.

As discussed also in the introduction, the manual way to solve this problem is typically to locate tables on a regular grid, starting from one angle of the available area and locating each new table at the given minimum distance. The manual layout is defined here by imposing a regular grid starting from the top corner of the available area (shown on the plots of Figure 5). The manual approach could locate 30 tables (red dots in the first plot of Figure 5. Our optimization method could locate 36 tables (red dots in the second plot of Figure 5). Having 6 more tables increases the capacity for customers of 20\%: this can make a significant difference in terms of daily profits of the brewpub, without impacting the compliance to the local social distancing rules (here assumed to be 3m) and the safety of the customers.

It is clear from this test (and others in [3]) that the optimal placement of tables is often not straightforward, and that the usage

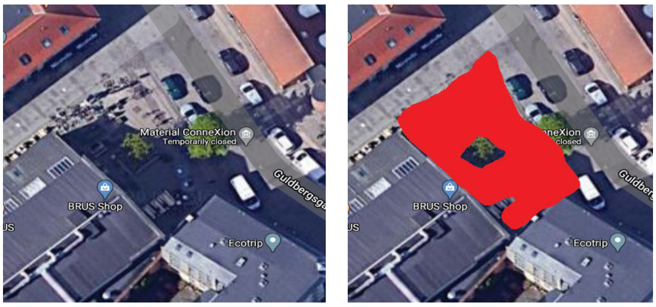

Figure 4: Example of optimization of table placement for the outside serving area of a brewpub in Denmark. The available area for placing tables is highlighted in red in the second plot.

of optimization methods can significantly increase the capacity of a restaurant to fit customers, and thus its daily revenue.

3.2 **Fitting a given number of tables while minimizing virus spread**

Another variant of the problem consists in fitting a fixed amount of tables in the area, while maximizing the safety of the customers. or these tests, we want fit as many tables as in the manual solution (see first plot of Figure 5) but in a safer way. In other words, with respect to the previous test, the focus is now shifted from maximizing the number of tables to minimizing the virus spread, while still fulfilling the country regulations on minimum table distances.

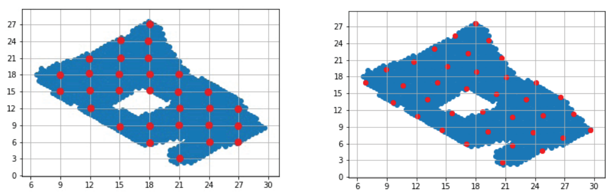

Figure 5: Example of optimization of table placement for the outside area of ``Brus" brewpub in Denmark. Tables are placed following a manual approach (based on a super-imposed 3m $\times$ 3m regular grid) in the first plot. The second plot shows the optimized placement of tables at a minimum of 3m distance, using an optimization tool: 6 more tables can be located.

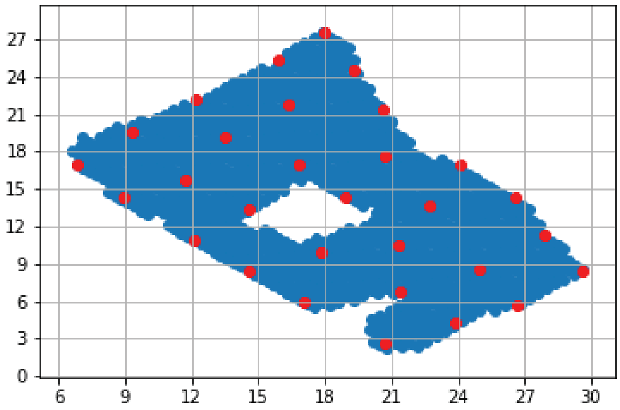

Figure 6: Minimizing virus spread for a fixed number of facilities smaller than the maximum capacity.

We will therefore still require the minimum distance of 3m between tables, but we impose to locate 30 tables in the Danish brewpub (while have seen that up to 36 tables could fit). Using the virus-spread function defined in [3] to measure the actual risk of infection between tables. The resulting layout is shown in Figure 6. Also this layout confirms that the positions on the borders are safer than those in the middle of the area.

Beach umbrellas

Another possible application of our optimization method is the optimal location of umbrellas on a beach. This summer, many seaside activities have been challenged by the COVID-19 restrictions, and owners of seaside areas faced difficulties in secure their income while ensuring safety. In countries like Italy, many beach areas are managed by privates who rent beach facilities (such as umbrellas, sunbeds, chairs, etc) to customers. Due to COVID-19, social distance limitations also apply in defining the position of the umbrellas on the beach. Using optimization methods instead of relaying on manual layouts, can have a big impact also in this case, allowing one to fit more customers while not compromising on their safety. For example, we considered a real case from beach ``Bagni Alberoni", located in Venice, Italy. The first plot of Figure 7 shows the actual layout designed by the beach owners to cope with the required 4m minimum distance between umbrellas. This solution locates 203 umbrellas in the available area. We gave the same area (in blue in the second plot of Figure 7) on input to our optimizer, together with the minimum distance of 4m with the goal of fitting as many umbrellas as possible within the given limitations. Our optimizer was able to fit 211 umbrellas --- having 8 more umbrellas to rent out over the whole summer season, can make a significant economical impact for local business.

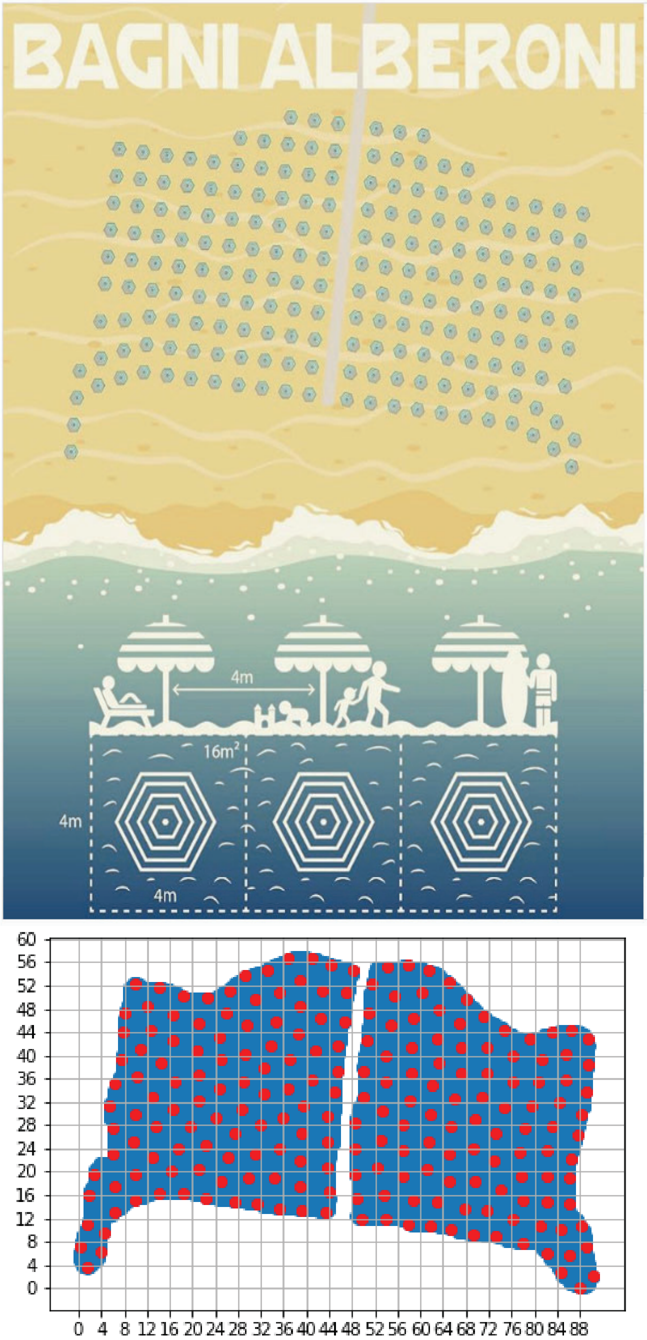

Figure 7: Example of a real case---the Venice beach ``Bagni Alberoni”. Beach umbrellas must ensure a minimum distance of 4m (center-to-center). The manual solution actually implemented (top) allocates 203 beach umbrellas, while the optimized one (bottom) is able to fit 211 beach umbrellas using a less-regular pattern.

Conclusions

We have studied the problem of locating facilities in a given area, subject to social distancing constraints as those arising at the time of COVID-19. We have proposed an analogy between this problem and the one of locating wind turbines in an offshore area, which allowed us to apply state-of-the-art solution approaches for the latter problem to produce optimized facility layouts. Real examples have shown that improved solutions can be obtained with less-regular (but more efficient) layout patterns than those typically found manually.

References

[1] M. Fischetti, M. Monaci, Proximity search heuristics for wind farm optimal layout, Journal of Heuristics 22 (4) (2016) 459-474.

[2] M. Fischetti, J. R. Kristoffersen, T. Hjort, M. Monaci, D. Pisinger, Vattenfall optimizes oshore wind farm design, INFORMS Journal On Applied Analytics 50 (1) (2020) 80-94.

[3] M. Fischetti, M. Fischetti, J. Stoustrup, Mathematical optimization for social distancing, technical report, University of Padova (2020).

Martina Fischetti is lead engineer in Operations Research in Vattenfall BU Wind. She has a PhD degree in Operations Research from DTU (industrial PhD in collaboration with Vattenfall), with focus on the usage of advanced mathematical optimization in the design of offshore wind farms.

By Niels-Christian Fink Bagger, Evelien van der Hurk, David Pisinger

Splitting Social Networks to Limit the Spread of COVID-19

The COVID-19 pandemic caused many countries to lock down large parts of society by establishing different policy measures aimed at limiting total contacts as a mean to decrease the spread of the virus. People are working from home, cultural events are cancelled and people can only meet up in small gatherings, meaning that the contacts between individuals have significantly dropped. Although the need to prevent the uncontrolled spread of COVID-19 is clear [1], the set of successful policy measures that enable this is not.

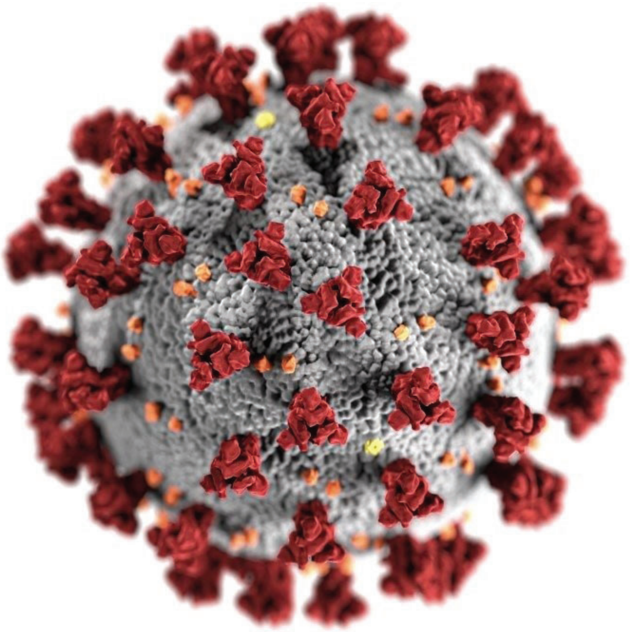

Figure 1: A Coronavirus

Which limitation policies should we introduce? How strict should they be? Can we increase the number of contacts that we find important, like family and friends? Which people at a workplace need to work from home, and how often? Which of the students at a course can be present at a lecture? These are some of the questions that will be investigated by the research project *Finding the "new normal": the power of distinct contacts* at the Technical University of Denmark funded by the Independent Research Fund Denmark. The objective of the project is to provide decision support tools to enable policymakers to evaluate, compare, and suggest new contact limitation measures with the objective of maximizing the number of contacts allowed while staying within a target acceptable disease spread.

Methodology

One of the most common ways to simulate a disease is via compartmental models such as the SEIR model [2]. These models try to predict things such as the spread of the disease, the total number of infected, the duration of the epidemic, or the effect of different policy measures. In the SEIR model the population is assigned to four compartments labeled, S, E, I and R (Susceptible, Exposed, Infectious and Removed). The susceptible compartment contains the individuals that have not been infected by the virus. A susceptible individual can transition to the exposed compartment when it gets into contact with an infectious individual and contracts the disease. The

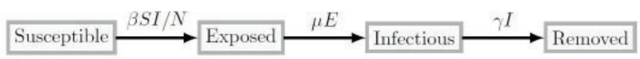

Figure 2: The SEIR (Susceptible, Exposed, Infectious and Removed) compartmental model

exposed compartment contains the individuals that have been infected, but cannot yet infect other individuals. The infectious compartment contains the individuals that have been infected and can infect others. The removed compartment contains the individuals that have been infected by the virus and have either recovered from the disease or died. An individual can transition from S to E, from E to I and lastly from I to R. The transitions between the compartments are illustrated in Figure 2. In Figure 2, the boxes illustrate the compartments and the arrows illustrate the transitions between compartments. The notation above the arrows describe the transition rate between the compartments at a time step. It is assumed that the transition rate from S to E at a given time step is $dS/dt = -\beta SI/N$, where S, I and N are the number of susceptible individuals, infectious individuals and the total population respectively. β is the probability that the individuals in the population gets into close contact, multiplied by the probability of getting infected from such a contact. It is assumed that the probability of an individual to transition from E to I is μ, so if E is the number of exposed individuals at a time step, then μE of those individuals will transition to the infectious compartment. This means that an individual is expected to spend $\mu{-1}$ time steps in the exposed compartment. This is referred to as the incubation time. The transition from I to R is given by the probability γ, which means that the individuals are expected to be infectious for $\gamma{-1}$ time steps, and γI individuals will transition from I to R at a given time step where I is the number of infectious individuals in that time step. The system can be described using the following differential equations:

$$\frac{dS}{dt} = -\frac{\beta SI}{N}$$
$$\frac{dE}{dt} = \frac{\beta SI}{N} - \mu E$$
$$\frac{dI}{dt} = \mu E - \gamma I$$
$$\frac{dR}{dt} = \gamma I$$

Some of the challenges regarding these compartmental models is the difficulty to evaluate individual based measures, such as self-quarantining. When an individual gets infected and discovers that they are infected, they are likely to go into self-quarantine and get tested for COVID-19. If an individual gets tested positive for COVID-19, it is attempted to trace the individuals that the infected has been in close contact with for the last couple of weeks.

The close contacts of the infected individual are then also likely to go into self-quarantine and get tested for COVID-19. As many limitation policies are limiting the contact between people it makes sense to look at the spread of the virus in network models [3], where every individual is represented by a node, and the contacts between individuals are represented by edges. Such models makes it easier to track individuals that have been in contact with each other for the self-quarantining measures. In the network models we still use the compartmental models by assigning the label to each node of the compartment that the corresponding individual is in. While a node is labeled E (or I) then, at every time step, the node is randomly assigned the label I (or R) with probability μ (or γ). The networks we consider are dynamic as individuals do not meet in every time step, so the edge between two individuals are only active in time steps where they meet, i.e., the virus can only be transmitted from an infectious node to a susceptible node, in a given time step, if the nodes share an edge, none of them are in self-quarantine, and the edge is active. This means that the transitions from compartment S to E, are based on the structure of the network and the state of the nodes, where by state of a node, we mean which label is assigned and whether or not it is self-quarantined. Most research in the spread of a virus in networks are considering static networks, but as most social contacts are varying over time, dynamic networks are more interesting.

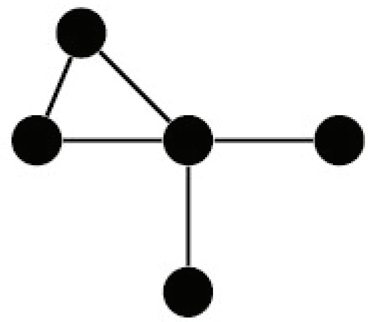

Figure 3: Example of a network of individuals (nodes) and the contacts (edges).

In our research we consider dynamic hyper graphs to estimate the contact patterns between individuals. We use the hyper graphs to evaluate the spread in a population within a fixed time horizon. Just like in regular networks, every node in the hyper graph corresponds to an individual in the population that we are considering. Instead of edges, we have *hyper edges*. A hyper edge is not restricted to connect at most two nodes, but can connect any number of nodes. Every hyper edge in the hyper graph corresponds to a group of individuals meeting up one or more times during the time horizon, where the hyper edge is active, i.e., like in regular networks, virus can only be transmitted via the hyper edge at every time step that the hyper edge is active. As an example, a course at a university could be represented by a hyper edge, where the nodes that are contained in the hyper edge corresponds to the students that are following the course and where the hyper edge is active every time there is a lecture in the course.

The reason we consider hyper edges, is that some limitation policies can be seen as splitting up hyper edges. As an example, if we have a limitation policy that at most 10 individuals are allowed to meet at any time, that corresponds to limit the hyper edges to contain at most 10 nodes, so for any hyper edge containing more than 10 nodes, we split the hyper edge into multiple hyper edges, such that each hyper edge contain at most 10 nodes. In this way, we only need the information about the hyper graph itself, whereas, if we used a regular graph, then we would need additional information about how the graph was constructed, to be able to evaluate the changes to the graph due to the limitation policies.

When limitation policies are introduced, the split of the hyper edges is uncertain as we cannot know how the graph is decomposed, e.g., if there is a limit on social gatherings for 10 people, we cannot know which 10 people different individuals meet up with at social gatherings. Some places we can have impact on how the graph is decomposed, this could be schools, universities or work places, and these are the cases we are initially considering in the project. One of our current focus areas is to investigate the best way to split a given hyper graph, with a given set of restrictions that limits the size of the hyper edges, where we are currently looking into minimizing distinct contacts.

Distinct Contacts

Given a hyper graph *G = (V,H)*, we define the set of distinct contacts as the set:

$$\left\{(u,v) \in V \times V \,\middle|\, \exists h \in H : \{u,v\} \subseteq V_h\right\}$$

Given a maximum size $M_h \in \mathbb{Z}$ for each $h \in H$, the objective is minimize the set of distinct contacts by replacing each hyper edge $h \in H$ with a new set of hyper edges such that the number of hyper edges in this new set does not exceed $\left\lceil \frac{|V_h|}{M_h} \right\rceil$ and the number of nodes in each of the new hyper edges does not exceed M_h, while ensuring that all the nodes *h* are contained in the union of the nodes of the new hyper edges. Each new set of hyper edges that replaces an existing hyper $h \in H$ edge is called a split of the hyper edge *h*. The total set of all the new hyper edges is called a *split* of the hyper graph. In the following definitions we state what a feasible split is.

Definition 1. Consider a hyper graph G = (V,H), a set of maximum sizes $\{M_h\}_{h \in H}$ *and a set of hyper edges H'. The set H' is a feasible split of H if, for every hyper edge ,* $h \in H$ *there exists a set of hyper edges* $h \in H$ *such that H'' where the following conditions are fulfilled:*

$$\begin{aligned}
&|H''| \leq \left\lceil \frac{|V_h|}{M_h} \right\rceil && (1)\\
&|V_{h'}| \leq M_h, \qquad \forall h' \in H'' && (2)\\
&\bigcup_{h' \in H''} V_{h'} = V_h && (3)\\
&V_{h'} \bigcap V_{h''} = \emptyset, \quad \forall h' \in H'', h'' \in H'' \setminus \{h'\} && (4)
\end{aligned}$$

If the conditions are fulfilled in Definition 1, then *H''* is a feasible split of *h*, and *H'* is a *feasible split* of *H*. We use the terms, *feasible split* of *H* and *feasible split of G* interchangeably as we only consider splitting up the graph by the hyper edges.

Currently in the project, we are working on models and algorithms to split the hyper edges, such that the number of distinct contacts are minimized, and then we intend to run the simulations on the resulting hyper graphs to evaluate the effect of decreasing the number of distinct contacts. Note that the number of distinct contacts for a node in the hyper graph, corresponds to the degree of a node in a regular graph. Early research has considered nodes with a high degree as *super spreaders*, but it has later been challenged that other measures are more important when trying to predict the outbreak [3]. Later in the project, we intend to look into these and other measure to use for splitting the hyper graphs, e.g., by identifying community structures [4]. Such measurements include vitality and centrality, e.g., node vitality, closeness centrality and betweenness centrality [4]. We hope to include these measurements into our splitting algorithms so we can provide tools for, e.g., employers or university administrations to help them in limiting epidemics while interfering as little as possible in the daily activities.

References

[1] Brendon Sen-Crowe, Mark McKenney, and Adel Elkbuli. Social distancing during the covid-19 pandemic: Staying home save lives. The American journal of emergency medicine, 2020.

[2] William Ogilvy Kermack and Anderson G McKendrick. A contribution to the mathematical theory of epidemics. Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character, 115(772):700– 721, 1927.

[3] Romualdo Pastor-Satorras, Claudio Castellano, Piet Van Mieghem, and Alessandro Vespignani. Epidemic processes in complex networks. Reviews of modern physics, 87(3):925, 2015.

[4] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. Physical review E, 69(2):026113, 2004.

Niels-Christian Fink Bagger is a Postdoctoral Researcher at the Technical University of Denmark, where he is working on the project Finding the "new normal": the power of distinct contacts, together with Associate Professor Evelien van der Hurk and Professor David Pisinger.

By Linnea Ingmar

Modelling Diversity of Solutions

While the majority of the literature within optimisation focuses on finding a *single* optimal solution to a given optimisation problem, my master thesis project [1,2] investigates the problem of finding a *set* of (near-optimal) solutions that illustrate some form of *diversity*.[1]

Motivation

The reason why we started working on this problem was an industrial collaboration within the research group at Monash University in Australia – where I was working during the thesis project – concerning the layout design of a chemical plant. This is an extremely complex problem involving finding 3D-positions for pieces of equipment and routing connecting pipes, subject to construction, operation, maintenance and safety constraints [3]. The Monash researchers were developing an automatic tool for finding optimal plant layout designs, with the aim of replacing the previously manual design process. However, it was hard to capture all the important aspects of the problem with an objective function, and therefore the plant engineers were not completely satisfied with the designs suggested by the tool. It seemed what was optimal in theory, was not optimal in practice.

This inspired us to work on diversity of solutions. Perhaps there are many plant designs with the same, or very similar, objective value, so why not present a set of candidate designs to the plant engineers?

As with most good ideas, we were not the first to come up with this. In fact, there has been a lot of previous work on diversity of solutions in many contexts [4,5]. The key advantage of our work compared to previous approaches is that it is not tied to a specific problem or solver. Instead, it is a generic framework where diverse solution problems are specified in a high-level constraint modelling language.

1 I wish to give credit to my eminent supervisors Prof. Maria Garcia de la Banda, Prof. Peter J. Stuckey, and Dr. Guido Tack, at Monash University, Australia.

So far so much talk about diversity. Now for the obvious question: what *is* diversity?

Modelling Diversity

The simple answer is: what diversity is depends on your problem. Consider again the plant layout problem. It would be useless for the plant engineers to compare two designs, where in one of them each piece of equipment has just moved by a few millimetres compared to in the other design. From an engineer's perspective, those two designs are identical. Instead, changing for example the *relative* positions of the equipment (e.g., swapping positions), or moving some specific piece of equipment by several meters, would be much more interesting for a plant engineer. Therefore, central to our modelling framework are user-defined *distance measures* between solutions. These distance measures are what defines diversity for your problem. The user specifies the distance measures directly in the constraint model, depending on their interest and need. Of course, some generic measures of distance, such as Euclidian distance, can be used as building blocks when specifying them.

In addition to distance measures, our modelling framework consists of components for *constraining the optimality* of the solutions (e.g., in percentage of optimality), *constraining distances* (e.g., setting some threshold value on a distance measure), and *combining* several distance measures. The underlying problem that one specifies with these components is finding k maximally diverse solutions to the original constraint satisfaction (CSP) or optimisation (COP) problem, where the number of solutions k is a parameter to the problem.

I mentioned earlier that we implemented the framework in a high-level constraint modelling language. This language is called MiniZinc[2] [6,7], and can be used to model CSPs and

2 Available from https://www.minizinc.org/.

COPs. The high-level MiniZinc model is compiled into a lower level language called FlatZinc, which is understood by a wide range of solvers. This means that you can try many different solvers without having to change your model of the problem. At the time of writing, the diversity framework is not part of the official MiniZinc distribution, but hopefully it will be in the future.

Solving Diversity Problems

We investigate a few different methods for finding diverse solutions. It is possible to solve the problem exactly (that is, to find the set of *provably* maximally diverse solutions) by encoding the problem into a new COP and essentially solving *k* copies of the original problem simultaneously. As you might guess, this method does not scale well, so we use a number of approximation methods: an *iterative, greedy* algorithm, a *hybrid* between the exact and the iterative method, and a *post-hoc* method that first generates a big set of solutions and then filters out the *k* most diverse.

Evaluation

We illustrate the usability of the framework on a number of problems, some artificial and some real-world. Among the solution methods we can see no clear winner across all benchmarks, so probably it depends on the application which one is best.

What about the plant layout design problem? Figure 1 shows three optimal designs (projected into 2D), with different positions of the equipment relative to the big pipe rack in the middle. Unfortunately, my work on the other parts of the project never left me time to learn the 3D CAD visualisation tool we used. As a compensation for the resulting poor image quality of Figure 1, I will instead end with a picture of a wallaby (Figure 2). Thanks for reading!

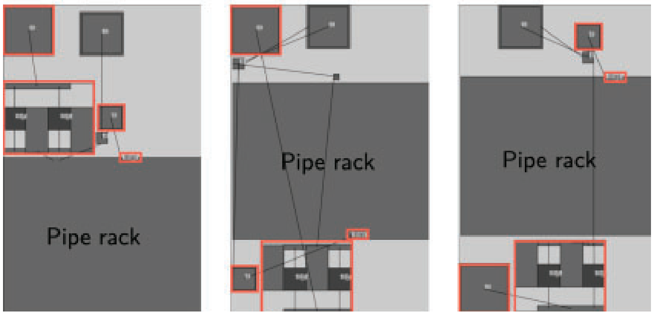

Figure 1: Three diverse optimal solutions to a small instance of the plant layout problem. Equipment encircled with red have different relative position to the pipe rack across the solutions.

References

1. Ingmar, L. “Modelling Diversity of Solutions,” MSc thesis, Uppsala University, Uppsala, 2019.

2. Ingmar, L., Garcia de la Banda, M., Stuckey, P.J., Tack, G. ”Modelling Diversity of Solutions,” in proceedings of the 34th AAAI Conference on Artificial Intelligence, 2020.

3. Belov, G., Czauderna, T., Garcia de la Banda, M., Klapperstück, M., Senthooran, I., Smith, M., Wybrow, M., Wallace, M. “Process plant layout optimization: Equipment allocation,” in proceedings of the 24th International Conference on Principles and Practice of Constraint Programming, 2018.

4. Hebrard, E., Hnich, B., O’Sullivan, B., Walsh, T. “Finding diverse and similar solutions in constraint programming,” in proceedings of the 20th AAAI Conference on Artificial Intelligence, 2005.

5. Petit, T., Trapp, A.C. “Finding diverse solutions of high quality to constraint opti- mization problems,” in proceedings of the 24th International Joint Conference on Artificial Intelligence, 2015.

6. N. Nethercote, P.J. Stuckey, R. Becket, S. Brand, G.J. Duck, and G. Tack. “MiniZinc: Towards a standard CP modelling language,” in proceedings of the 13th International Conference on Principles and Practice of Constraint Programming, 2007.

7. Peter J. Stuckey, Thibaut Feydy, Andreas Schutt, Guido Tack, and Julien Fischer. “The MiniZinc challenge 2008-2013,” AI Magazine, 35(2):55–60, 2014.

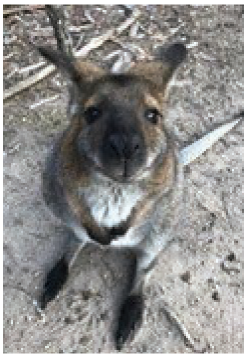

Figure 2: A wallaby, which belongs to the same diverse class of mammals – marsupials - as the kangaroo.

Linnea Ingmar is a PhD student at KTH Royal Institute of Technology, Sweden. Her research interests are combinatorial optimisation, programming languages, compilers, and machine learning.

Email: lingmar@kth.se

By Peter Emil Tybirk

DORS Price Winner Article

A primer on reinforcement learning for combinatorial optimization

Introduction

For most operations research researchers and practitioners, it is enticing to ride along the machine learning (ML) hype-train and leverage some of the exciting new techniques which has risen to fame during the last five to ten years. Combinatorial optimization (CO) lies at the heart of many of the problems which OR researchers study and posing the question of how machine learning techniques can help to solve combinatorial optimization problems is therefore natural.

There are several paradigms in machine learning that, in different ways, can be useful when solving combinatorial optimization problems. The omnipresent supervised learning paradigm is probably the most widely used and studied. Supervised learning be utilized in many ways when solving combinatorial optimization problems. Supervised learning is in essence function approximation, i.e. to 'learn' a mapping $f{:}X \rightarrow Y$ between inputs $x \in X$ and outputs $y \in Y$.

In the most ambitious setting, X would contain all conceivable instances of a certain type of combinatorial optimization problems and Y the corresponding optimal solutions, and f would be a function that given any $x \in X$ could produce an optimal solution. It is unrealistic to expect such an approach to have success in general as the existence of f in a space of functions which we can both define and approximate well is a pretty tough precondition. The most promising class of functions to look for f in may often be in one of the many variants of graph neural networks (GNNs). GNNs are a class of neural networks designed to take graphs as inputs and output e.g. a class, a number or another graph. Decent attempts at utilizing supervised learning in this ambitious setting have been made for several classes of combinatorial optimization problems [1].

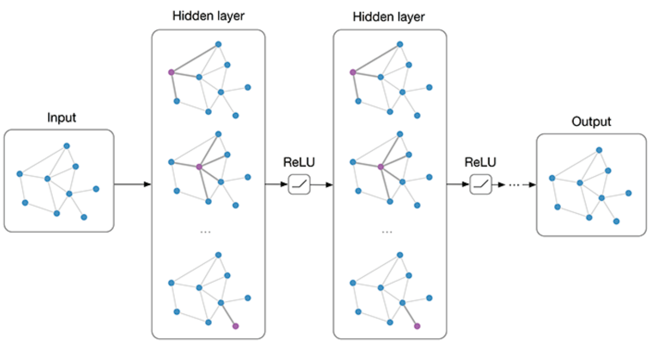

Figure 1: Example graph neural network

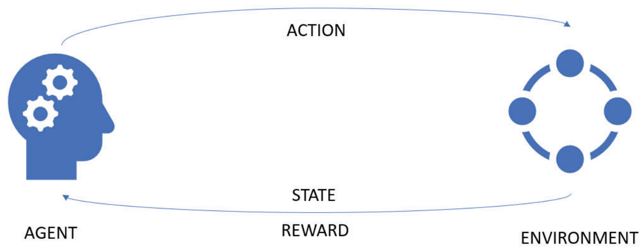

Figure 2: State/action framework

An approach with more immediate potential is to use ML techniques as subroutines in a CO algorithm. This can be framed in the state/action framework known from Markov Decision Processes (MDPs). The framework is applicable both for exact and heuristic algorithms. For instance, an action could be the choice of variable to branch on in the context branch and bound for Mixed Integer Linear Programs. Or an action could be which local-search strategy or permutation to apply in a heuristic algorithm. Supervised learning techniques are typically used to improve an already existing algorithm for a problem, whereas *reinforcement learning* typically is an 'end-to-end' approach which requires less a priori knowledge about the specific problem.

Reinforcement learning

In contrast to supervised learning, reinforcement learning does not require training examples with optimal solutions to learn from. Reinforcement learning in general revolves around an *agent* trying to maximize a notion of *reward* in an *environment* by choosing the right *actions*.

The most famous application of RL techniques is probably the groundbreaking *AlphaGo* and *AlphaZero* which tackled the board games Go and Chess and set new standards in both games. It was the first time an algorithm bested the best humans in Go – a game previously thought to be out of reach for computers for many years to come. Board games such as Chess and Go bear many similarities with combinatorial optimization problems (some CO problems can even be transformed into games [2]), which makes applying the same techniques appealing.

There are several ways to frame combinatorial optimization problems in a reinforcement learning context – depending on how one chooses to represent the four core components, the *state space*, the *environment*, the *action space*, and the *reward structure*. Defining these three properly central aspects is the key to utilize reinforcement learning for combinatorial optimization. In the following sections are some examples of how these core components can be defined for CO problems.

Pros of RL for CO	Cons of RL for CO
•Can learn to exploit latent structure in data without human instruction •Generalizable approach across many CO problems •Environments in CO problems can be simulated and have many of the right traits to be treated as RL problems •Exciting research area!	•Computationally demanding in training phase •Notoriously difficult to train agents - sensitive to hyperparameters and reward structure •Defining the right reward structure, state and action space is not straightforward

The environment

The environment encompasses different instances of the same class of combinatorial optimization problem. During training, the agent should see many different instances of a given class of CO problems to be able to generalize and learn the common structure such problems share.

An important design choice is how to define what constitutes an *episode* (a 'round of training'). A popular framework is the solution *construction* framework - to build a solution from scratch and stop once a feasible solution has been reached. In a sense, RL would in this case constitute a greedy algorithm where the agent iteratively constructs a solution using a *learned* greedy measure, which is continuously improved during training (if all goes well!).

Another framework is the solution *mutation* framework where an action turns one feasible solution into another one with the goal of finding the optimal solution (the agent of course has no general way to know that an optimal solution is reached unless it hits a known lower bound). An episode in this case could for instance stop once a certain number of actions have been made without improvement to the best seen objective value.

Not all CO problems are well-suited for both frameworks. For instance, if finding a feasible solution is in itself difficult, neither approach can be expected to succeed.

The action space

The action space inherently depends on the nature of the optimization problem and the choice of framework.

In the construction framework it is often straightforward to define the action space – it could for instance consist of all the unused edges in a graph, and an action could be to include an edge.

In the mutation framework an action can have different levels of granularity. The purest and most ambitious approach is to go with as granular actions as possible, i.e. to let actions be very small mutations of a feasible solution. In the travelling salesman problem (TSP), this could for instance be a single 2-opt exchange.

Another approach is to let actions consist of existing heuristics for mutating a solution, e.g. different variants of a local search, different perturbations to escape local minima, etc. Reinforcement learning can in this context be seen as a kind of hyperheuristic, where the goal of the agent is to learn to compose (meta)-heuristics depending on the state of the optimization problem. An immediate advantage of this approach is that it can help identify good combinations of heuristics, see figure 3.

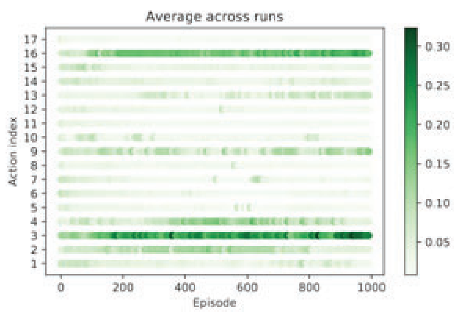

Figure 3: Proportion of actions taking by different agents in each episode - the agents learn to prefer a combination of action 3, 9 and 16

In the 'fine-grained' setup, the agent is on the other hand not constrained by the inherent limitations in the available heuristics but faces a much more daunting task requiring more 'deep' choices which reflect the full state of the current solution – if this approach is implemented successfully it has larger potential as the agent is not constrained by the available heuristics, but actually making it work is a much more difficult task.

The state space

Designing the state space, one should include all information necessary for the agent to make the right actions. However, including too much information may introduce unnecessary complexity and increase the computational effort needed.

The construction and the granular mutation approach require a complex state representation to represent the current solution and enable the agent to make 'informed choices'. In the TSP, the agent would for instance need to know distance between each of the nodes in the graph. An immediate consequence of a complex state-space is that the state-action value function which needs to be approximated in most RL algorithms would also be complex – probably a variant of a GNN.

By contrast, the coarser mutation based approach requires less expressivity in the state, and thus simpler function approximations will also suffice. In this case, the state could contain some history of actions, short and long-term memory, the current objective value and other key statistics about the current solution.

The reward structure

The final piece in the puzzle is the reward structure. In the construction approach, the objective value in the constructed solution immediately comes to mind. One important thing to consider is a scaling of the rewards such that the reward does not increase or decrease merely by scaling all values in the problem, as the reward needs to be similar across different instances of the same problem. One simple way of achieving this would be to scale the reward by some value related to the problem to achieve a 'relative performance' (RP), for instance the relative performance at time t may be defined as

$$RP_t = \frac{objective_t - LB}{UB - LB}$$

Where LB is some lower bound on the objective value, and UB some upper bound. In the construction approach, a reward would only be given at the end of an episode once a solution is reached. In the mutation-based approach a similar approach can be used, e.g. by giving a reward each time the relative performance is improved, and perhaps giving out more reward when the relative performance is already good, e.g.

$$reward_t = \frac{RP_{t-1} - RP_t}{|RP_{t-1}|}$$

Note that this is assuming we are considering a minimization problem – similar structure can easily be defined for a maximization problem.

Final thoughts

The above sections described the basic building blocks needed to apply reinforcement learning for combinatorial optimization. The rest of the work is picking your favorite RL algorithm and function approximator, which should of course match the state representation. Luckily there exists a lot of well-documented software packages (especially in Python) which provide RL and/or function algorithms off-the-shelf, so you just need to pick your favorite CO problem, define these building blocks, and have fun watching the agents train!

There has not yet been a breakthrough in RL for CO of the same magnitude as AlphaZero, but the approach has shown good potential on a range of problems, and I strongly believe that a breakthrough is coming. In my master's thesis, I proposed and used the mutation-based approach to discover effective heuristics for the fixed-charge transportation problem – in the end a new state-of-the-art heuristic was developed, not relying on RL per se, but heavily inspired by the strategies found by RL agents.

If this article aroused your curiosity, you can read more and see an example implementation of reinforcement learning for the fixed charge transportation problem here: https://github.com/Tybirk/RL-FCTP (feel free to reach out with questions!) and find papers and code on other implementations of reinforcement learning for solving combinatorial optimization problems here: https://paperswithcode.com/task/combinatorial-optimization.

References

[1] Yoshua Bengio, Andrea Lodi, Antoine Prouvost, Machine learning for combinatorial optimization: A methodological tour d'horizon, European Journal of Operational Research, 2020

[2] Ryuiang Xu, Karl Lieberherr, Learning Self-Game-Play Agents for Combinatorial Optimization Problems

Peter Emil Tybirk works as a Machine Learning Engineer at Mindway AI where he develops algorithms for early detection of problematic gambling behavior. He has a MSc degree in Mathematics-Economics from Aarhus and was the recipient of the 2019 DORS price for the thesis Reinforcement Learning for Combinatorial Optimization with Application to the Fixed Charge Transportation Problem.

By Julia Pahl

DORS Price 2020 Announcement

DORS - Danish Operations Research Society – hereby ask for candidates for Denmark's best master thesis project in Operations Research

The invitation is open for all and not just members of DORS - so please share!

A thesis can only be submitted by the supervisor and must comply with the following:

- The thesis is written at a Danish university and should involve Operations Research
- The thesis is written in Danish or English
- The thesis has been defended in 2020

The committee will put emphasis on analyzing and solving practical problems when selecting the best thesis

To enlist a thesis as candidate for the DORS prize 2020, the supervisor should send an email no later than 31st of January 2021 to secretary@dorsnet.dk with the following:

- The thesis (attached as pdf)
- A description of why the thesis should win the prize (max 1 page), written by the supervisor. (attached as pdf)
- The grade of the thesis
- Email addresses and phone numbers of the authors of the thesis and the supervisor

After the thesis is submitted to DORS the following occurs:

- DORS sets up a committee of three professionals.
- In the end of February 2021, the committee writes a confirmation email to the supervisors and authors of the submitted theses.
- In the middle of March 2021, the committee appoints a winner and informs all the participants
- The prize is awarded at the General Assembly of DORS in April 2021. The prize is of 5000 DKK to share between the authors of the thesis.
- To receive the money, the authors are required to write a short article to "ORbit" regarding the thesis.

Please write secretary@dorsnet.dk if you have any questions.

By Rasmus Riber

Quayside Planning in Container Terminals

An Applied Study of Quay Crane Scheduling

Purpose

This thesis addresses operational aspects in quay crane scheduling (QCS). At field studies conducted in three terminals, it was observed that plans from commercial QCS planning tools were discarded straight away and adjusted manually to account for operational aspects. The purpose of this thesis is to improve quay side planning and operation through better quay crane scheduling. On a high level the hypothesis is that better QCS can reduce vessel makespan via better utilisation of QCs and thereby increase terminal capacity. The thesis focuses on two main topics: 1) Improve QCS by bridging gaps between state-of-the-art planning models and terminal operations. 2) Reveal insights on key factors for the QCS. In this article for ORbit, only a subset of content addressing 1) is presented due to brevity and confidentiality matters. The project is conducted in collaboration with Portchain. Portchain is a Danish start-up company focused on solving the hardest planning problems in shipping. Their container terminal solution is currently used by several terminals globally.

Introduction

Around 80% of the global trade volume and 70% of global trade value is carried by sea and handled by sea ports. Containerised maritime trade has historically been growing 3-4% per year to account for 18% of volume and 60% of value for all maritime trade in 2018 ([UNC18], [UNC19], [Dep20]). Port calls have become a key bottleneck for containerized shipping due to increasing vessel sizes the last two decades. Since mid 00's the largest container vessels have grown from around 10,000 to 24,000 TEU [Cou20]. The increased size implies increased container throughput per port call and increased complexity of terminal planning. A rough analysis on a recent study reveals that major liner services typically spend 10-20% of the total round trip time in port ([CS19]). Vessel fuel consumption increases with a cubist relation to speed, hence reduced time in port will i.a. enable reduced consumption of costly and polluting bunker fuel by sailing slower. It is estimated that the average utilisation of major container terminals is around 70%. At the same time, the direct berth rate in major terminals is only at 80-90% and the average waiting time to berth is 7-9 hours ([Kno15], [Moo17], [Res19]). The paradox of having waiting time while not having full utilisation is caused by several factors, with inefficiencies in terminal planning and operations accounting for 86.1% of the liner scheduling unreliability, as observed in figure 1. 50% of the vessels are delayed by 12 hours or more. For ultra large container vessels, capacity of +15.000 TEU, the waiting time is often higher due to fewer quay cranes at the terminals being capable and available of handling the ultra large vessels ([Tec15],[Boy19],[PS19]).

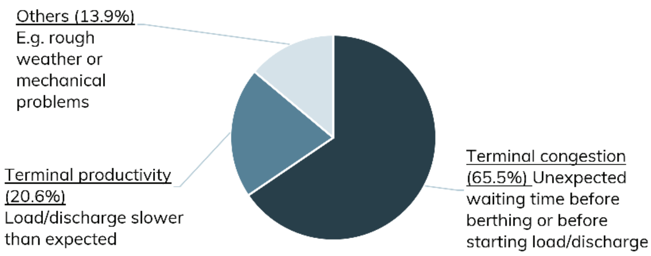

Figure 1: "Causes for liner shipping delays [Not06]"

Better planning supported by data driven decision tools can increase terminal utilization and reduce waiting time and delays. In other words, the terminal capacity can be increased without investing in additional heavy assets. It is estimated that better planning can increase capacity by 10% in most terminals [Boy19]. Quay crane scheduling (QCS) is a key link in the planning chain and is considered to be the hardest to solve from a computational point of view ([BM10],[BM15]).

Operational Quayside Planning

Quayside planning consists of berth allocation, quay crane assignment and quay crane scheduling chronologically depicted in figure 2. The berth allocation defines when and where the vessel is berthed along the quay. It is planned up to two weeks ahead of vessel arrival and is based on depth requirements, quay crane requirements and container positioning on the yard. The quay crane assignment (QCA) defines which quay cranes will be assigned to the vessel in each time window. It is planned up to one week ahead of vessel arrival and is based on the expected productivity, the total moves, blue collar gang working hours and the berthing. The quay crane scheduling defines the bays each QC should operate on the vessel at a given time. This is planned 0-48 hours ahead of vessel arrival and is based on the vessel discharge and loading plans, the quay crane allocation and the yard allocation. This thesis focuses on quay crane scheduling (QCS), which is considered the hardest problem from a computational point of view ([BM10],[BM15]).

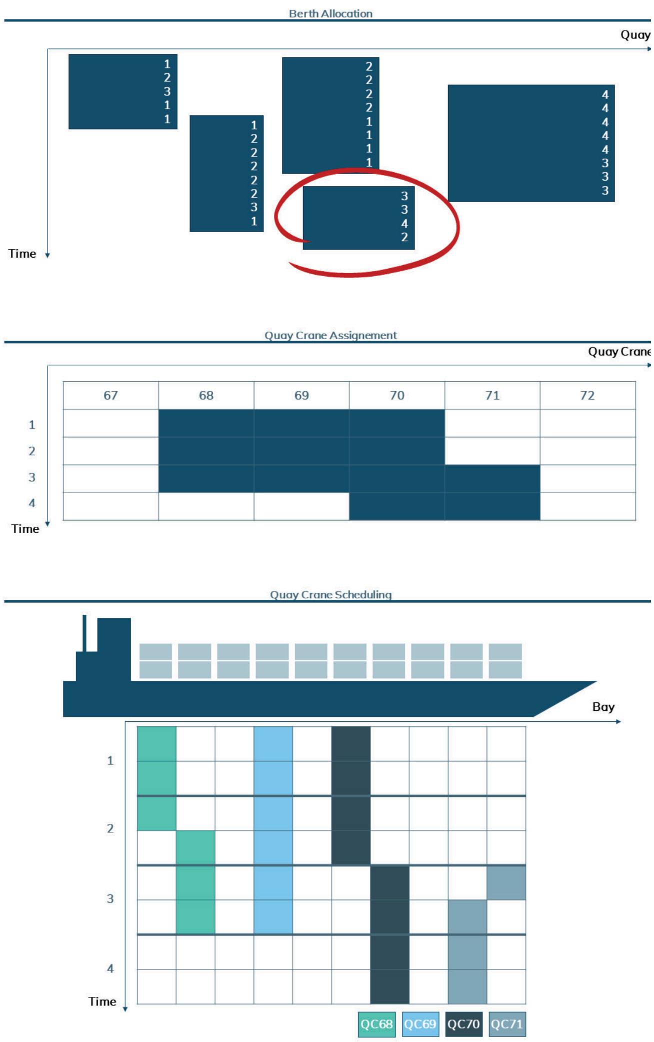

Figure 2: "Berth allocation (each square represents a vessel), QCA and QCS."

Planning KPIs

The main QCS KPI for terminals is to deliver the promised makespan for a vessel. The planned makespan is usually based on a contractual agreement of the average berth moves per hour between the terminal and the carrier. The shortest possible makespan increases quay capacity for the terminal. Furthermore, it allows vessels to slow-steam which yields significant bunker cost savings for the carrier. However, planning the shortest possible makespan is of high risk to delays due to the uncertain nature of physical operations but also due to simplification of the plans. The QCS planners operate after three KPIs ranked in order of importance:

The reliability KPI covers a) how often the planned makespan is effectuated and b) how many times the plan needs to be changed during operation. Planning to minimise the makespan is the second KPI. The third KPI is to minimise QC moves between bays as it increases safety risk and decreases the productivity. However, moving QCs between bays increases flexibility and thereby reliability.

QCS Modelling Branches

The academic QCS literature can be segmented in three main branches based on the discretization level of jobs: bay models, group models and single container models. Figure 4 shows an example where a total of 10 container moves is discretized

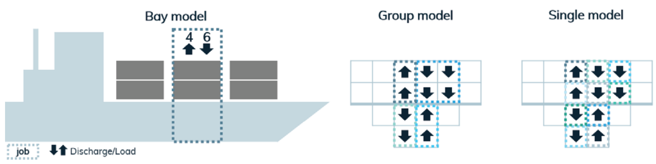

Figure 4: "Bay, group and single container modelling of a generic 10 move bay."

in one, four and 10 jobs for each branch respectively. The majority of the literature formulates the QCS as bay or group models, however recent research proposes single container models. Several articles point out Kim and Park [KP04] to bring the initial formulation for group level modelling. Bay and group models have achieved far better computational speed than single container models ([CLG14],[Msa+18],[STB19]).

In this study, the group modelling approach is identified to be best suited based on literature results, experimental analysis and a fit-for-purpose assessment rooted in data gathering from site visits at three container terminals.

Operational Rich Planning vs Basic Planning

A mathematical model F covering the key operational aspects of the QCS is developed based on inputs from planners and managers at the terminal visits. F is assumed to reflect the actual operation. The hypothesis that a model FB covering the basics only can generate as good a plan as the operational and more complex model F is tested. The deviation in makespan from FB plans in operational environment compared to F for benchmark suites A, B and C, is listed as (I) in table 1. (II) lists the results when planning FB including superstructures and booming. (III) lists the results when planning FB including dual cycling. A plan by FB lasts 96 minutes (4.5%) longer on average than a plan by F, when tested on benchmark suite C. Accounting for dual cycling (III) yields the smallest deviation to the makespan by F. Accounting for superstructure and booming (II) has limited effect on the decrease in makespan deviation.

		A	B	C
(I)	Makespan deviation [Minutes] ([%])	36 (3.6)	60 (3.9)	96 (4.5)
(II)	Makespan deviation [Minutes] ([%])	26 (2.6)	47 (3.1)	80 (3.8)
(III)	Makespan deviation [Minutes] ([%])	16 (1.5)	28 (1.8)	40 (1.9)

Table 1: “Makespan deviation between FB and F in operational environment. 1 CME = 2 minutes.”

Summing up the main findings regarding the operational rich planning it is found that: (1) FB estimates a similar makespan as F in less computational time. (2) Plans by F can be operationally executed with lower makespan than plans by FB. As stated in figure 1 slower loading and discharging than expected accounts for 20% of the liner scheduling unreliability. This is in accordance with the analysis performed which found up to 4.0% gap between the estimated makespan by FB and its actual makespan in operational environment. It is further found that planning with F enables up to 4.5% shorter makespan than planning with FB. Lastly, it is worth noting that the ultra large container vessels are not covered by benchmark suit A, B and C by Meisel, hence even higher deviations could potentially be found.

Conclusion

This thesis addresses operational aspects in quay crane scheduling (QCS). At field studies conducted in three terminals it was observed that plans from commercial QCS planning tools were discarded straight away and adjusted manually to account for operational aspects. A benchmarking of state-of-the-art QCS MIP modelling branches led to the selection of a group based unidirectional MIP formulation as base model FB. A revised rich MIP formulation F is proposed to account for operational aspects which have not been covered in previous studies e.g. dual cycling. F and FB estimate the makespan within 3% difference. Execution of the FB is by simulation found to be 4.5% longer than execution of F, equaling more than one and a half hour for vessels with 20 bays and 6000 total moves. The substantial difference justifies the complexity of the proposed operational richness. The operational richness increases computational time from on average 4s to 100s for large instances. A proposed matheuristic, based on adding restrictive constraints to the model, enables the rich model F to be solved in 7.0 seconds with 0.1% optimality loss on average. The model can serve as basis for an operational optimisation tool for terminal planners given its short computation time and the flexibility offered by the rich model.

References

[BM10] - Christian Bierwirth and Frank Meisel. “A survey of berth allocation and quay crane scheduling problems in container terminals”. In: European Journal of Operational Research 202.3 (2010), pages 615–627.

[BM15] - Christian Bierwirth and Frank Meisel. “A follow-up survey of berth allocation and quay crane scheduling problems in container terminals”. In: European Journal of Operational Research 244.3 (2015), pages 675–689.

[Boy19] - Magnus Boye. Dansk startups algoritmer optimerer driften af containerterminaler.

2019. url: https://pro.ing.dk/datatech/artikel/dansk-startups-algoritmer-optimerer-driften-af- containerterminaler-3222. (accessed: 20.03.2020).

[CLG14] - Jiang Hang Chen, Der-Horng Lee, and Mark Goh. “An effective mathematical formulation for the unidirectional cluster-based quay crane scheduling problem”. In: European Journal of Operational Research 232.1 (2014), pages 198–208.

[Cou20] - World Shipping Council. Container Ship Design. 2020. url: http://www.worldshipping.org/about-the-industry/liner-ships/container-ship-design. (accessed: 20.03.2020).

[CS19] - Claude Comtois and Brian Slack. Ship Turnaround Times in Port: Comparative

Analysis of Ocean Container Carriers. CIRRELT, Centre interuniversitaire de recherche sur les réseaux d'entreprise …, 2019.

[Dep20] - Statista Research Department. Container Shipping Worlwide. Dossier, 2020.

[KP04] - Kap Hwan Kim and Young-Man Park. "A crane scheduling method for port container terminals". In: European Journal of operational research 156.3 (2004), pages 752–768.

[Kno15] - Greg Knowler. Mega-ship delays less than overall average at EU ports, data shows. 2015. url: https://www.joc.com/port-news/mega-vessels-delays-less-overall-average-eu ports-data-shows_20150722.html. (accessed: 20.03.2020).

[Moo17] - Turloch Mooney. Wait times for ships to berth lengthen at major prorts. 2017. url: https://www.joc.com/port-news/port-productivity/shipsizes-grow- wait- times- berth-lengthen_20171209.html. (accessed: 20.03.2020).

[Msa+18] - Mohamed Kais Msakni et al. "Quay crane scheduling with vessel stability".

In: Transportation research procedia 30 (2018), pages 60–69.

[Not06] - Theo E Notteboom. "The time factor in liner shipping services". In: Maritime Economics & Logistics 8.1 (2006), pages 19–39.

[PS19] - Nam Kyu Park and Sang Cheol Suh. "Tendency toward mega containerships and the constraints of container terminals". In: Journal of Marine Science and Engineering 7.5 (2019), page 131.

[Res19] - Drewry Maritime Research. Global Container Terminal Operators Annual Review and Forecast 2019. Drewry, 2019.

[STB19] - Defeng Sun, Lixin Tang, and Roberto Baldacci. "A Benders decomposition based

framework for solving quay crane scheduling problems". In: European Journal of Operational Research 273.2 (2019), pages 504–515.

[Tec15] - Port Technology. Average Ship Waiting Times at Global Ports. 2015. url:

https://www.porttechnology.org/news/average_ship_waiting_times_at_global_ports/. (accessed: 20.03.2020).

[UNC18] - UNCTAD. Review of Maritime Transport 2018. United Nations Publications, 2018.

[UNC19] - UNCTAD. Review of Maritime Transport 2019. United Nations Publications, 2019.

Rasmus Riber graduated as honours student from DTU with an M.Sc. in Transport and Logistics Optimisation this summer.

By Dennis S. Holm, Rasmus Ø. Mikkelsen, and Thomas Stidsen

Winning the International Timetabling Competition 2019

1 Introduction

The International Timetabling Competition 2019 (ITC2019) is the fourth such competition. The competitions are held to encourage research within the field. The first two competitions of 2002 and 2007 focused on simplified university course timetabling, whereas the 2007 competition also included exam timetabling. The competition of 2011 focused on high school timetabling. The ITC2011 was groundbreaking because it presented the high school timetabling problem with a generalized formulation, such that it can be used at high schools worldwide. The ITC2019 has the same groundbreaking aspect, but concerning university timetabling. The ITC2019 was presented with a generalized university course timetabling model, including student sectioning (assigning students to classes given the course assignment). The data used in the competition was real data from 10 universities from 5 continents. The competition started in November 2018 and finished one year later, November 2019, with the final results presented in an online award ceremony on September 2, 2020.

2 The competition

The competition was presented at the International Conference on the Practice and Theory of Automated Timetabling (PATAT 2018) in Vienna, Austria. It was presented with a generalized problem description and a few test data sets of different sizes. The problem formulated is a university course timetabling problem with student sectioning. The aim is to find an optimal assignment of times, rooms, and students to events (classes) related to a set of courses.

The formulation is derived from data from institutions worldwide, and thus it must include many different aspects of timetabling. Some constraints are typical for all institutions, such as assigning all classes to a time and a room and forbidding room double-booking. Other restrictions do not show too often, like discouraging the number of breaks on a day or the limitation of working hours for an instructor. So-called distribution constraints define the restrictions between the scheduling of classes. There have been described 19 different distribution constraints that can appear to be soft or hard.

The competition was run through a website (www.itc2019.org), where competitors could validate and upload solutions to the competition data instances. For each instance, the competitors were ranked against each other, and a score was given according to ranking. Thus the algorithm used is not handed over to the organizers, and neither were there any restrictions on the computer power used to generate the solutions.

Three groups of data instances were released during the competition (early, middle, and late). The early instances were released on November 15, 2018, the middle September 18, 2019, and the late instances on November 8, 2019. The final score was based on the ranking of the solutions uploaded on the deadline date (November 18, 2019), where the later instances weighted higher than the earlier. See Table 1 for the scoring scheme. If two solutions are tied, the average points for the positions are granted for all solutions. If no solution is uploaded, the team is awarded zero points.

3 The competition problem

The generalized model is used to schedule a whole semester for an institution, usually between 13 and 21 weeks. The formulation should then be able to model various time policies as institutions differ in this aspect. Some institutions might have strict rules about when a class can start; for example, every hour (or quarter/half past), other institutions might be more relaxed, allowing classes to start whenever. The generalized problem handles the time differences by splitting a day to be into 288 timeslots (5 minutes each) from midnight till midnight for all the instances. Each class has defined a set of time patterns that can be used to schedule the class. The time pat-

Position	Early	Middle	Late
1.	10	15	25
2.	7	11	18
3.	5	8	15
4.	3	6	12
5.	2	4	10
6.	1	3	8
7.	-	2	6
8.	-	1	4
9.	-	-	2
10.	-	-	1

Table 1: Points awarded per instance. Ranking is based on the computation of points in the F1 championship

tern consists of a set of weeks, a set of days, a start time, a duration, and a penalty for using the time pattern. Thus, if a class can only be scheduled Tuesday every quarter past an hour, the class's time patterns will reflect that. Other differences shown in the data was the ability to schedule classes on single/multiple days, in the first/last half of the semester, in odd/even weeks, and on evenings/Saturdays.

Most classes must also be assigned a room. Each class has given a set of available rooms associated with a penalty. The rooms' availability and penalty are generated from various characters, such as the capacity, location, features, equipment, and preference. On top of that, the room can have unavailable times that must be respected by the solution method.

The remaining restrictions of the timetable are given by the distribution constraints, which were briefly mentioned earlier. The distribution constraints explain the feasibility/penalty between scheduled classes. All distribution constraints consider a set of classes that must be scheduled according to the distribution constraint type. All distribution constraint types are shown in Table 2 and can both appear as a soft or hard constraint. Most of the distribution constraint types are self-explanatory. The ones that are not will be explained here. The `Overlap` and `SameAttendees` are closely related. The `Overlap` says that the classes cannot overlap in time; the `SameAttendees` specifies that there should also be enough time for students to get from one class to the next. The travel distance between rooms is given in the data. If two classes are scheduled on the same day, it should be such that the travel time between the rooms is less than the time between the classes.

The `WorkDay(S)` limits the time between the first start time and last ending time on any day, which must not be more than `S` timeslots. The `MinGap(G)` says that there must be at least `G` timeslots between classes scheduled on the same day. The `MaxDay(D)` regulates the number of different days used, which must not exceed `D` days. The `MaxDayLoad(S)` restricts the number of timeslots on any day to be no more than `S`. The `MaxBreaks(R,S)` defines a block to be classes scheduled on the same day with at most `S` timeslots between them. The constraint says that there must be no more than `R` breaks between blocks on any day. The `MaxBlock(M,S)` also defines a block like `MaxBreaks(R,S)`. This constraint limits the length of a block to be at most `M` time slots.

Having the available time patterns and rooms given for the classes and the distribution constraints restricting the scheduling between classes, we can create a feasible timetable.

Constraint	Opposite	Pairs
SameStart		✓
SameTime	DifferentTime	✓
SameDays	DifferentDays	✓
SameWeeks	DifferentWeeks	✓
SameRoom	DifferentRoom	✓
Overlap	NotOverlap	✓
SameAttendees		✓
Precedence		✓
WorkDay(S)		✓
MinGap(G)		✓
MaxDays(D)		
MaxDayLoad(S)		
MaxBreaks(R,S)		
MaxBlock(M,S)		

Table 2: The different types of distribution constraints and whether they can be evaluated in pairs or not.

But some universities also consider the conflicts in the students' class assignments. The student sectioning must be performed according to the structure of the courses that the student must attend. A course may have a complex structure of classes: lectures, recitations, and/or laboratory. The course may have different configurations where each student must attend one of the configurations. Each configuration may consist of one or more subparts where the student must attend one class from each subpart of the configuration. An example course structure is shown in Figure 1. The students must be sectioned in classes such that the limitation of the classes is not exceeded. A class may also have a parent class, which means that students attending the class must also attend its parent class. Whenever a student is assigned two classes that overlap, either by the time pattern overlapping or the room assignment is too far apart, we have a student conflict. Each student conflict is considered a soft constraint and penalized in the objective function.

The reader might see that we now have four objective types:

```xml
<course id="ME 263">
  <config id="1"> <!-- Lec-Rec configuration, not linked -->
    <subpart id="1_Lecture"> <!-- Lecture subpart -->
      <class id="Lec1" limit="100"/>
      <class id="Lec2" limit="100"/>
    </subpart>
    <subpart id="2_Recitation"> <!-- Recitation subpart -->
      <class id="Rec1" limit="50"/>
      <class id="Rec2" limit="50"/>
      <class id="Rec3" limit="50"/>
      <class id="Rec4" limit="50"/>
    </subpart>
  </config>
  <config id="2"> <!-- Lec-Rec-Lab configuration, linked -->
    <subpart id="3_Lecture"> <!-- Lecture subpart -->
      <class id="Lec3" limit="100"/>
      <class id="Lec4" limit="100"/>
    </subpart>
    <subpart id="4_Recitation"> <!-- Recitation subpart -->
      <class id="Rec5" parent="Lec3" limit="50"/>
      <class id="Rec6" parent="Lec3" limit="50"/>
      <class id="Rec7" parent="Lec4" limit="50"/>
      <class id="Rec8" parent="Lec4" limit="50"/>
```

Figure 1: Example of hierarchical course structure with its XML specification (Müller et al., 2018).

the room penalties, the time penalties, the soft distribution constraints, and the student conflicts. Each institution may prioritize these differently; thus, the data instances contain a weighting between the four objective types.

4 Our solution approach

Our solution approach is strongly MIP based. We use MIPs to find initial solutions, we use MIPs to improve solutions, and we use MIPs to prove optimality. But first, as mentioned earlier, the data is real-world data, and the problem with real-world data is that real people have entered the data at some point, and real people tend not to have a full overview of their data. The first step is thus to reduce the data instances by finding redundant information.

4.1 Reducing the data sets

The data contains redundant distribution constraints. This includes soft constraints with penalty zero, constraints considering only one class, and constraints of the same type where one set of classes is a subset of the other. These are removed.

To reduce the number of variables in the MIP, we reduce the number of available rooms and times for the classes. We do so by constructing a conflict graph where a vertex corresponds to a class-time assignment, and an edge is added if two vertices cannot both be chosen in a feasible solution. Some classes have only one available time, which means that it is a fixed time. Any neighbor of a fixed-time vertex is forbidden, and the neighbors can be removed from the graph. Also, considering a clique in the conflict graph containing all vertices of a class c_1 and other vertices of classes c_i $(i \neq 1)$, we know that one of those vertices of class c_1 must be used in a feasible solution; the c_i vertices can thus be removed. All removed vertices represent class-time assignments that cannot be used in any feasible solution, which means that the times can be removed from the classes. An equal procedure is done for the class-room conflict graph.

4. 2 The MIP

The MIP considers the main binary decision variables $x_{c,t,r}$, which is 1 if a class c is scheduled in room r at time t and 0 otherwise. We also define the binary auxiliary variables $y_{c,t}$ and $w_{c,r}$ which exactly correspond to the vertices of the conflict graph from section 4.1. If the problem includes student sectioning, the MIP also contains binary decision variables $e_{s,c}$, which is 1 if a student is attending a class and 0 otherwise.

We use the conflict graphs to model the distribution constraints that are pairwise comparable according to Table 2. A clique cover models the hard constraints. The `SameAttendees` distribution constraint requires an additional conflict graph on $x_{c,t,r}$ to model the time-room overlap. For the soft constraints, we use conflict graphs as well. The edge weight corresponds to the penalty of breaking the soft constraint(s) defining the edge. The soft constraints are then modeled as a star cover.

The student sectioning is modeled like the `SameAttendees` constraint as they are equal in structure.

4. 3 Initial solutions

We have two methods for creating initial solutions. The first splits the problem into two parts and is called the 'Two-Stage Constructive Algorithm' (2SCA). The first stage constructs a feasible timetable and the second part adds the student sectioning. The first step of 2SCA solves a MIP that does not consider any soft constraints and no students. Stage two solves a MIP that assigns students to classes given the timetable from stage one.

The other initial solution method is the 'Three-Stage Constructive Algorithm' (3SCA). The first stage of 3SCA is to assign times to classes. The second stage is to assign rooms given the times from stage one. This is not always possible, and thus stage one must be repeated, disallowing any previously found time assignments. When a feasible time and room assignment has been found, stage three solves a MIP that assigns students to classes given their time and room assignments from stage one and two.

4. 4 Fix-and-Optimize

To improve the initial solutions, we introduce the Fix-and-Optimize matheuristic. The Fix-and-Optimize splits the set of main decision variables into the two sets F and U. We can then optimize a subproblem where the variables of F are fixed and those of U are not. The selection of the sets strongly affects the success of the matheuristic. If the size of U is large, the problem can become too hard to solve in reasonable time. On the other hand, if U is small, it is unlikely the subproblem can be used to find improving solutions. We define different heuristics to select U and update the size of U dynamically.

4.5 Computational setup

When the data was released, we started by reducing the data files as described. When we have created the reduced files, we started the 2SCA and 3SCA (Section 4.3), which were set to create a pool of initial solutions. In parallel, we started building the full MIP (Section 4.2) and several Fix-and-Optimize (Section 4.4) algorithms with different neighborhoods. The MIP is focusing on improving the lower bound while the Fix-and-Optimize algorithms focus on producing improved solutions that are passed to the MIP to help reduce the branch-and-bound tree. The Fix-and-Optimize algorithms consider different neighborhoods and regularly reset to the best-known solution. If enough time passes with no improvement in the best known solution, the Fix-and-Optimize algorithms enter diversification mode. They individually start from a new initial solution from the initial solutions pool, they no longer share best-known solutions, and they choose a neighborhood at random. When an improving solution is found, they return to the normal strategy. The whole procedure terminates when the time limit is reached, or the MIP proves optimality.

5 The Results

On September 2, 2020, an online award ceremony was held, where the final results were published. The final ranking is shown in Table 3. The ceremony also revealed the methods used by our competitors. The second place, Rappos et al. (HEIG-VD, Switzerland), also used a MIP approach. They combined their MIP model with a local search procedure. The third place, Gashi et al. (University of Prishtina, Kosovo), used simulated annealing. The fourth place, Karim Er-rhaimini (Ministère de l'éducation nationale, France), used a forest growth metaheuristic. The fifth place, Lemos et al. (Universidade de Lisboa, Portugal), used MaxSAT combined with local search. At the time of the competition deadline, 15 teams had uploaded one or more solutions to the instances (including test instances). Even though the competition has officially ended, the website is still running, and a live scoreboard is kept updated. The number of teams has now increased to 20, and the site has 263 registered users from 57 countries. If you would like to read more about the competition, problem, or would like to give it a try to solve the problem, visit www.itc2019.org. If you would like to read more about our team and see our current solutions and lower bounds on the ITC2019 instances visit www.dsumsoftware.com/itc2019

Pos.	Team	Early	Middle	Late	Total
1	Holm et al.	99	150	240	**489**
2	Rappos et al.	72	94	156	**322**
3	Gashi et al.	41	85	147	**273**
4	Er-rhaimini	36	71	145	**252**
5	Lemos et al.	17	32	30	**79**

Table 3: Competition results. The score of the ITC2019 Early, Middle, and Late instances, and the Total Points.

References

Müller, T., Rudová, H., and Müllerová, Z. (2018). University course timetabling and International Timetabling Competition 2019. In Burke, E. K., Di Gaspero, L., McCollum, B., Musliu, N., and Özcan, E., editors, Proceedings of the 12th International Conference on the Practice and Theory of Automated Timetabling (PATAT-2018), pages 5–31.

Dennis Søren Holm is a Ph.D. student at DTU Management. His Ph.D. project relates to the application of Operations Research techniques for Strategic University Timetabling. Link: https://dsumsoftware.com/strategic-university-timetabling/

Rasmus Ørnstrup Mikkelsen is an industrial Ph.D. student employed by MaCom A/S and enrolled at DTU Management. His Ph.D. project concerns the application of Operations Research techniques for Quality Timetable Recovery. Link: https://dsumsoftware.com/quality-timetable-recovery/

Thomas Jacob Riis Stidsen is an Associate Professor of Operations Research at DTU Management. His research includes educational timetabling, scheduling, and resource planning.

By Daniele Gammelli

A Machine Learning Approach to Shared Mobility Demand Prediction

1 Introduction

Being able to understand and predict demand is essential in the planning and decision-making processes of any given transport service, allowing service providers to make decisions coherently with user behavior and needs. Having reliable models of demand is especially relevant in shared transport modes, such as car-sharing and bike-sharing, where the high volatility of demand and the flexibility of supply modalities (e.g., infinitely many possible collocations of a car-sharing fleet) require that decisions be made in strong accordance with user needs. If, for instance, we consider the bike sharing scenario, service providers face a great variety of complex decisions on how to satisfy user demand. To name a few, concrete choices must be made for what concerns capacity positioning (i.e., where to deploy the service), capacity planning (i.e., dimensioning the fleet size), rebalancing (i.e., where and when to reallocate idle supply), and expansion planning (i.e., if and how to expand the reach of the service).

Demand modeling uses statistical methods to capture user demand behavior based on recorded historical data. However, historical transport service data is usually highly dependent on historical supply offered by the provider itself. In particular, supply represents an upper limit on our ability to observe realizations of the true demand. For example, if we have data about a bike-sharing service with 100 bikes available, we might observe a usage (i.e., demand) of 100 bikes even though the actual demand might have potentially been higher. This leads to a situation in which historical data is in fact representing a biased, or censored, version of the underlying demand pattern in which we are truly interested. More importantly, using censored data to build demand models will, as a natural consequence, produce a biased estimate of demand and an inaccurate understanding of user needs, which will ultimately result in non-optimal operational decisions for the service provider.

To address these problems, we propose a general approach for building models that are aware of the supply-censoring issue, and which are ultimately more reliable in reflecting user behavior. To this end, we formulate Censored Gaussian Processes as a model of user demand. Using real-world datasets from Donkey Republic, one of the major bike-sharing services in Copenhagen, Denmark, we pit this model against non-censored models (i.e. models ignoring the censoring problem) and analyze the conditions under which it is better capable of recovering true demand.

2 Methodology

In this Section, we incrementally describe the building blocks of our proposed censored models. First, we introduce several general concepts: likelihood-based training, Censored likelihoods and Gaussian Processes. Then, we combine these concepts by defining the Censored Gaussian Processes which we will be using for our real-world experiments.

2.1 Machine Learning and Likelihood-Based Training

Machine Learning (ML) is a branch of Artificial Intelligence (AI) that aims at discovering general principles underlying human learning, giving computers the ability to learn from experience. Current approaches in ML rely on extracting pattern and regularities from data. Considering the transportation sector as an example, the always increasing amount of available data makes ML models essential in identifying patterns of movements far better than even the most skilled human observer would be able to do. One of the most popular approaches to machine learning is known as Maximum Likelihood Estimation (MLE). The idea behind MLE is to estimate the parameters of any ML model by first defining the model from a probabilistic standpoint (e.g. parametrizing a specific probability distribution) and then estimate the parameters by maximizing the probability of the data given the model. In other words, MLE enables us to find the model which better describes our observable data (i.e. the model that gives higher probability to our data). In practice, MLE is solved as an optimization problem where the objective function is represented by the model's likelihood. Different likelihoods necessarily imply different objectives for our learned models. In this work, we will use the flexibility of likelihood-based learning to encode in our models awareness towards the censoring problem.

2.2 Censored Likelihood

As a reference point for developing our censored likelihood function, let us now elaborate on the likelihood function of the popular Tobit censored regression model, described in [3]. For each observation y_i, let y_{i^*} be the corresponding true value. For instance, in a shared transport demand setting, y_{i^*} is the true, latent demand for shared mobility, while y_i is the observed demand; if y_i is non-censored then $y_i = y_{i^*}$, otherwise y_i is censored so that $y_i < y_{i^*}$. We are also given binary censorship labels l_i, so that $l_i = 1$ if yi is censored and $l_i = 0$ otherwise (e.g., labels could be recovered by comparing observed demand to available supply).

Tobit parameterizes the dependency of y_{i^*} on explanatory features $\mathbf{x}_i$ through a linear relationship with parameters β and noise term ε_i, where all ε_i are independently and normally distributed with mean zero and variance σ^2, namely:

$$y_{i^*} = \beta^{\mathrm{T}}\mathbf{x}_i + \varepsilon_i, \qquad \varepsilon_i \sim \mathrm{N}(0, \sigma^2). \tag{1}$$

There are multiple variations of the Tobit model depending on where and when censoring arises. In this work, without loss of generality, we deal with upper censorship, also known as *Type I*, where y_i is upper-bounded by a given threshold y_u, so that:

$$y_i = \begin{cases} y_i^*, & \text{if } y_i^* < y_u \\ y_u, & \text{if } y_i^* \geq y_u \end{cases}. \tag{2}$$

The likelihood function in this case can be derived from Eqs. 1 and 2, as follows.

1. If $l_i = 0$, then y_i is non-censored and so its likelihood is: where φ is the standard Gaussian probability density function.

$$\frac{1}{\sigma}\phi\left(\frac{y_i - \beta^{\mathrm{T}}\mathbf{x}_i}{\sigma}\right), \tag{3}$$

2. Otherwise, i.e., if $l_i = 1$, then yi is censored and so its likelihood is:

$$1 - \Phi\left(\frac{y_i - \beta^{\mathrm{T}}\mathbf{x}_i}{\sigma}\right), \tag{4}$$

where Φ is the standard Gaussian cumulative density function.

Because all observations are assumed to be independent, their joint likelihood is:

$$\prod_i \left\{\frac{1}{\sigma}\phi\left(\frac{y_i - \beta^{\mathrm{T}}\mathbf{x}_i}{\sigma}\right)\right\}^{1-l_i} \left\{1 - \Phi\left(\frac{y_i - \beta^{\mathrm{T}}\mathbf{x}_i}{\sigma}\right)\right\}^{l_i}, \tag{5}$$

which is a function of β and σ. The intuition behind this likelihood form is that our model will essentially treat differently observations depending on whether these are effected by the censoring phenomenon (i.e. we know the true demand could have potentially been higher) or not (i.e. we know we can trust the observations as representing the desired true demand). Notice how the definition of this likelihood represents an attempt to encode supply-awareness into the implementation of demand prediction models.

2.3 Gaussian Processes

Gaussian Processes (GPs) [2] are an extremely powerful and flexible tool belonging to the field of probabilistic machine learning [1]. GPs have been applied successfully to both classification and regression tasks. Given a finite set of points for regression, there are typically infinitely many functions which fit the data, and GPs offer an elegant approach to this problem by assigning a probability to every possible function. Moreover, GPs implicitly adopt a full probabilistic approach, thus enabling the structured quantification of the confidence – or equivalently, the uncertainty – in the predictions of a GP model. This ease in uncertainty quantification is one of the principal reasons why we chose to use GPs for demand prediction in the shared mobility domain. Indeed, transport service providers are not only interested in more accurate demand models, but also, and maybe most importantly, wish to make operational decisions based on the measure with which the model is confident of its predictions. In practice, GPs express their uncertainty in predictions by updating their prior distribution (i.e. before observing the data) into a posterior distribution (i.e. distribution after having observed the data). As an example, Figures 1, 2 show the prior and the posterior on a simple regression task.

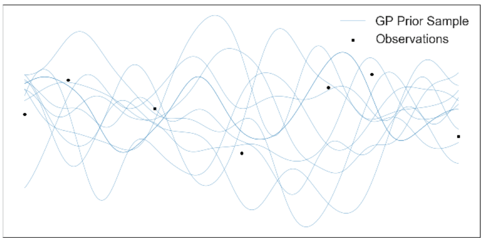

Figure 1: Samples from GP prior distribution

In what follows, we will combine the concepts of Censored Likelihood and Gaussian Processes to define Censored Gaussian Processes in an attempt to encode supply-awareness within the modeling flexibility of GPs.

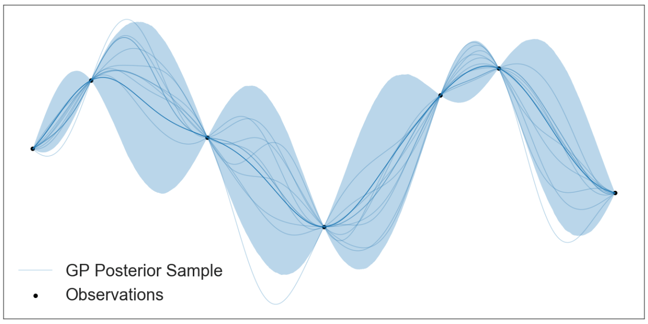

Figure 2: Samples (and corresponding confidence interval) from GP posterior distribution.

3 Bike-Sharing Demand Prediction

In this Section, we deal with the problem of building a demand prediction model for a bike-sharing system. Donkey Republic can be considered a hub-based service, meaning that the user of the service is not free to pick up or drop off a bike in any location, but is restricted to a certain number of virtual hubs around Copenhagen. Our objective is to model daily rental demand in the hub network. The given data consists of individual records of users renting and returning bikes in hubs during 379 days: from 1 March 2018 until 14 March 2019. Hence before modeling daily rentals, we aggregate the data both spatially and temporally. Spatially, 32 hubs were aggregated in three super-hubs by selecting three main service areas (such as the central station and main tourist attractions) and considering a 200 m radius around these points of interest (Figure 3). Temporally, the data at our disposal allowed us to retrieve the time-series of daily rental pickups regarding the three super-hubs, which will represent the target of our prediction model.

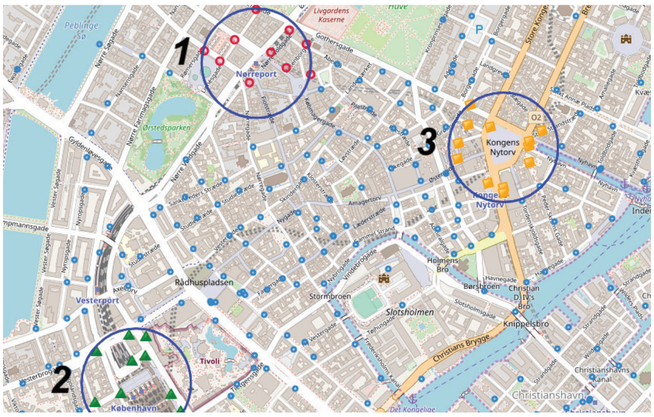

Figure 3: Map with: 1) marked locations of Donkey Republic hubs, 2) the three super-hubs in our experiments, as big circles around constituent hubs.

Ideally and before modeling, we would like to have access to the true bike-sharing demand, free of any real-world censorship. However, this ideal setting is impossible, as historical data records are necessarily censored intrinsically to some extent. Consequently and for the sake of experimentation, we assume that the given historical data represents true demand (which is what ideally we would like to predict). This further allows us to censor the data manually and examine the effects of such censorship.

We apply manual censorship to the time series of each super-hub in two stages. In the first stage, for each day *i* in *N* = *{1...379}*, we let $\delta_i \in \{0,1\}$ indicate whether at any moment during i there were no bikes available in the entire super-hub, and define accordingly the set of censored and non-censored observations:

$$N_c = \{i \in 1..379 : \delta_i = 1\}, \tag{6}$$

$$N_{nc} = \{i \in 1..379 : \delta_i = 0\} = N - N_c. \tag{7}$$

We then fix binary censorship labels as follows: l_i = *1* for $l_i = 1$ for $i \in N_c$ and $l_i = 0$ for $i \in N_{nc}$. The reason for doing so is that for every day in Nc, there was a moment with zero bike availability, and so there may have been additional demand, which the service could not satisfy and which was thus not recorded.

Having fixed the censorship labels, the second stage of censorship can be executed multiple times for different censorship intensities.

That is, given a censorship intensity $0 \le c \le 1$, we censor each observation for which $l_i = 1$ *to (1 − c)* of its original value.

3.1 Results

As introduced in previous sections, the focus of our experiments is the comparison between Censored and Non-Censored models in the estimation of true demand patterns. We thus compare three GP models:

(i) Non-Censored Gaussian Process (NCGP): represents the Gaussian Process model most commonly used in literature, i.e., with Gaussian observation likelihood. NCGP is trained on the entire dataset, consisting of both censored and non-censored observations, without discerning between them.

(ii) Non-Censored Gaussian Process, Aware of censorship (NCGP-A): functionally equivalent to NCGP, but uses information on censoring as a pre-processing step. That is, NCGP-A is trained only on non-censored points, thus avoiding exposure to a biased version of the true demand (because of censoring). This, however, comes at the cost of ignoring relevant information embedded in the censored data

(iii) Censored Gaussian Process (CGP): this model considers all observations – censored and non-censored – through the likelihood function defined in (Section 2.2).

This section presents results for the predictive models implemented on each of the three time-series with cross-validation. We now concentrate on the results for super-hub 1, as presented in Figures 4, 5 because they are representative also of the results for the other two super-hubs. The plots are a visual representation of Table 1 (below) and compare the performances of NCGP, NCGP-A and CGP for different censorship intensities. We discern between evaluating model performance on the entire dataset (consisting of both censored and non-censored observations) vs. only on non-censored observations.

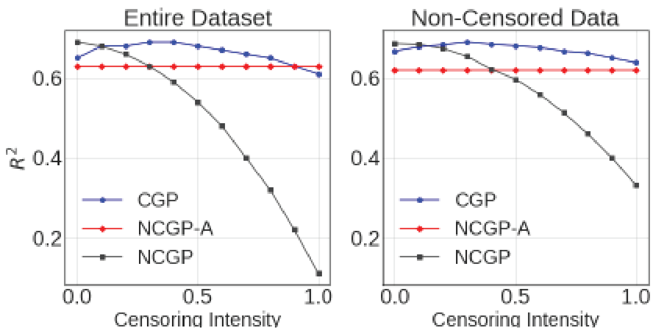

Figure 4: R^2 performance over varying censoring intensities

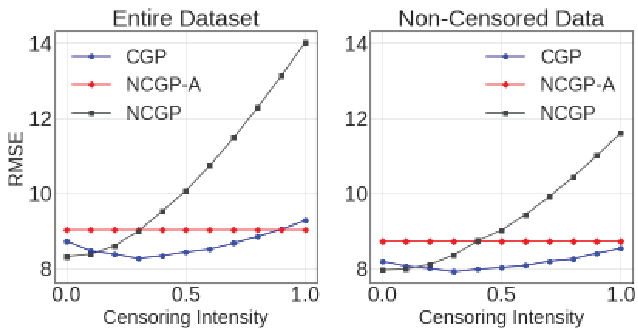

Figure 5: RMSE performance over varying censoring intensities

First, we compare the models that do not discard of any observations, namely, CGP and NCGP. Considering that a predictive model is better the more its RMSE is to close 0 and the more its R^2 is close to 1, the plots show that the two models are comparable under low degree of censoring. However, as the censorship intensifies, NCGP becomes strongly biased towards the censored observations, whereas CGP recovers the underlying demand much more consistently.

Next, we compare between NCGP-A and the CGP and see that NCGP-A achieves reasonable predictive accuracy, which is still mostly worse than the predictive accuracy of CGP. As it can be expected, NCGP-A accuracy depends highly on the extent to which observable data characterizes the full behavior of the latent function (in this case, true demand). Here, the percentage of points affected by censoring falls between 20% and 40% for all the three super-hubs, so that NCGP-A has acceptable observability over the true demand. Even so, CGP outperforms NCGP-A also on just non-censored data; this suggests that using a censored likelihood not only allows models to avoid predictive bias on censored data, but also allows consistent understanding of the data generating process, ultimately leading to increased performance also on observable data.

In conclusion, the non-parametric nature of Censored GP allows it to effectively exploit the concept of censoring, thus preventing censored observations from biasing the entire demand model. In other words, Censored GP is capable of activating censoring-awareness depending on data only.

4 Conclusions

Building a model for demand prediction naturally relies on extrapolating knowledge from historical data. This is usually done by implementing different types of regression models, to both explain past demand behavior and compute reliable predictions for the future – a fundamental building block for a great number decision making processes. However, we have shown how a reliable predictive model must take into consideration censoring, especially in those cases in which demand is implicitly limited by supply. More importantly, we stressed the fact that, in the context of shared transport demand modeling, there is a need for models which can deal with censoring in a meaningful way, rather than resorting to different data cleaning techniques.

To deal with the censoring problem, we have constructed models that incorporate a censored likelihood function within a flexible, non-parametric Gaussian Process (GP). We compare this model to commonly used GP models, which incorporate a standard Gaussian likelihood, through a series of experiments on real-world datasets. These experiments highlight how standard regression models are prone to return a biased model of demand under data censorship, whereas the proposed Censored GP model yields consistent predictions even under severe censorship. The experimental results thus confirm the importance of censoring in demand modeling, especially in the transport scenario where demand and supply are naturally interdependent. More generally, our results support the idea of building more knowledgeable models instead of using case-dependent data cleaning techniques. This can be done by feeding the demand models insights on how the demand patterns actually behave, so that the models can adjust automatically to the available data.

References

[1] Zoubin Ghahramani. Probabilistic machine learning and artificial intelligence. Tech. rep. University of Cambridge, 2015, pp.452-459.

[2] Carl Edward Rasmussen and Christopher K I Williams. Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press, 2005. isbn: 026218253X.

[3] James Tobin. “Estimation of Relationships for Limited Dependent Variables”. In: Econometrica 26.1 (1958), pp. 24– 36. issn: 00129682, 14680262.

Table 1: Model Performance for Super-hub 1

Censorship Intensity:		0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Entire Dataset												
RMSE	NCGP	**8.31**	**8.38**	8.6	8.99	9.52	10.07	10.74	11.48	12.29	13.14	14.03
	NCGP-A	9.03	9.03	9.03	9.03	9.03	9.03	9.03	9.03	9.03	**9.03**	**9.03**
	CGP	8.73	8.47	**8.38**	**8.26**	**8.34**	**8.44**	**8.51**	**8.67**	**8.85**	9.04	9.28
R^2	NCGP	**0.69**	**0.68**	0.66	0.63	0.59	0.54	0.48	0.40	0.32	0.22	0.11
	NCGP-A	0.63	0.63	0.63	0.63	0.63	0.63	0.63	0.63	0.63	**0.63**	**0.63**
	CGP	0.65	0.68	**0.68**	**0.69**	**0.69**	**0.68**	**0.67**	**0.66**	**0.65**	**0.63**	0.61
Non-Censored Data												
RMSE	NCGP	**7.96**	**7.99**	8.11	8.36	8.74	9.02	9.43	9.91	10.44	11.01	11.61
	NCGP-A	8.73	8.73	8.73	8.73	8.73	8.73	8.73	8.73	8.73	8.73	8.73
	CGP	8.19	8.06	**8.00**	**7.91**	**7.98**	**8.02**	**8.08**	**8.20**	**8.25**	**8.40**	**8.54**
R^2	NCGP	**0.69**	**0.68**	0.67	0.65	0.62	0.6	0.56	0.51	0.46	0.40	0.33
	NCGP-A	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62
	CGP	0.67	**0.68**	**0.68**	**0.69**	**0.68**	**0.68**	**0.68**	**0.67**	**0.66**	**0.65**	**0.64**

Daniele Gammelli is a Ph.D. Student at the Technical University of Denmark, where he works on machine learning models and optimization algorithms for understanding urban mobility. He received a MSc degree in Management Engineering from the University of Rome "La Sapienza", Italy. His research interests include machine learning, probabilistic graphical models, mathematical optimization, reinforcement learning, intelligent transportation systems, and urban mobility

By Joao Fonseca and David Koza

AOO 2021 Announcement

Save the date for AOO2021: May 3rd 2021

The Danish Operations Research Society (DORS) is pleased to announce the next Applications of Optimization conference (AOO2021), our annual workshop on applied optimization. It will take place on May 3rd at Industriens Hus in Copenhagen.

We are aware that the event may be subject to change depending on the further development of the Corona pandemic, and we will keep you updated through our online channels once we get closer to the date.

The workshop program is composed of four talks from OR experts and practitioners, and plenty of time for networking. We were able to secure the same panel of speakers that was scheduled for cancelled AOO2020, as follows:

- Anita Schöbel, Head of Fraunhofer Institute for Industrial Mathematics
- Henrik Caroe Bylling, Data Scientist at IKEA
- Marco Lübbecke, Professor of Operations Research at University of Aachen
- Niels-Christian Fink Bagger, Post-Doc at DTU Management

There are 5 free tickets available for DORS student members. If you are a DORS student member and interested in joining AOO 2020, please send us a motivated application (a few sentences) before April 15th in order to apply for one of the five free tickets.

We hope to see you all on May 3rd 2021!

The DORS board