Literatuuronderzoek

Ons probleem: Course Timetabling Problem (CTP)

Bij timetabling gaat het erom dat er een rooster wordt gemaakt waarin de vakken worden ingeroosterd op bepaalde dagen en timeslots. In zo’n rooster zitten de volgende elementen: studenten, lokalen, hoorcolleges, werkcolleges, practica en timelsots. Dit moet echter gebeuren op een manier waarop het rooster zich houdt aan belangrijke constraints.

Constraints waarop gelet wordt zijn:

* studenten kunnen niet meerdere vakken volgen binnen een timeslot (dus 1 vak per student per timeslot // geen overlappingen)
* iedere ingescrheven student moet in het lokaal passen (desnoods de groep opsplitsen in 2 aparte groepen)
* per lokaal kan er maar 1 evenement plaatsvinden
* \*\* een vak moet gelijkmatig verspreid worden over de week (bv maandag, woensdag en vrijdag)

**Comparison of SA and HC in the course timetabling problem – Kenekayoro Patrick**

[**http://www.academicjournals.org/journal/AJMCSR/article-full-text-pdf/EC495AE4687**](http://www.academicjournals.org/journal/AJMCSR/article-full-text-pdf/EC495AE4687)

Deze tekst vergelijkt de implementatie van hill climber tegenover simulated annealing.

Wat is Hill Climber precies? Bij Hill Climber worden er swaps gemaakt tussen elementen op het rooster, waarbij de swap uitgaat van een verbetering. Op deze manier wordt het rooster stap voor stap verbeterd. Echter is er ook een valkuil; bij Hill Climber kan het (meestal) voorkomen dat het swappen vastloopt in een *local maxima:* een staat waarin er geen optimale swaps meer zijn en het algoritme dus vastloopt. Dit komt o.a. ook zo omdat bij hill climber er geen stappen terug mogen worden genomen. Een oplossing is dan om het algoritme weer opnieuw randomly te runnen. Bij Simulated Annealing kunnen er echter wel stappen terug worden genomen waardoor het algoritme telkens door kan blijven itereren. Echter, is hier de valkuil dat dit er voor kan zorgen dat er swaps worden gemaakt met elementen die slechter zijn dan de instantie waarin het zich nu bevindt.

Er worden hier vier algoritmes getest:

* Simulated annealing (random) 🡪 slechte resultaten
* Simulated annealing (best) 🡪 vaak beter dan hill climbing
* Random restart hill climbing 🡪 beter dan regular hill climbing, en SA (want met een random restart wordt er weer een grote ruimte doorzocht)
* Regular hill climbing.

Echter, wordt het beste resultaat behaald door SA.

Bij geen een van de algoritmes was de penalty gelijk aan 0. Er waren bij elke dus wel maluspunten.

**A MULTI-PHASE APPROACH TO UNIVERSITY COURSE TIMETABLING - MINHAZ FAHIM ZIBRAN**

<https://www.uleth.ca/dspace/bitstream/handle/10133/633/zibran,%20minhaz.pdf;sequence=1>

Dit artikel splitst de solution approach op in vier fases. Interessant hier is dat er al meteen een onderscheid wordt gemaakt in dag-clusters. Zo wordt dinsdag en donderdag een cluster en maandag, woensdag en vrijdag.

* In de eerste fase worden de components (lectures, labs en tutorials) aangewezen aan instructors (professors, academic assistants en graduate students). 🡪 (dit is voor ons onderzoek echter irrelevant).
* In de tweede fase worden lectures toegewezen aan de dag-clusters.
* In de derde fase worden lectures van een enkele dag-cluster toegewezen aan timeslots die op die dag open zijn.
* In de vierde fase worden labs en tutorials aan dagen en beschikbare timeslots gekoppeld.

Misschien vergelijken met onze stappen?

**Solving the Course Scheduling Problem Using Simulated Annealing - E. Aycan and T. Ayav**[**http://web.iyte.edu.tr/~tolgaayav/papers/Aycan\_CourseSch.pdf**](http://web.iyte.edu.tr/~tolgaayav/papers/Aycan_CourseSch.pdf)

Om een SA algoritme te implementeren, wordt hier een neighborhood structure gedefinieerd. Dit wordt gezien als de key component voor elke SA methode.  
Er worden drie algormites beschreven:

* Simple Searching Neighborhood: It randomly chooses one activity and one slot. The chosen slot is assigned as the start time of the selected activity
* Swapping Neighborhoods: The second algo- rithm selects randomly two activities and swaps their start times
* Simple Searching and Swapping Neighborhoods: This neighborhood searching algorithm chooses randomly two activities and two slots. These two slots are assigned as the start times of the randomly selected activities

Simple Searching Neighborhood geeft hierin het beste resultaat.

**Simulated Annealing Cooling Schedules for the**  **School Timetabling Problem** - **David Abramson, Mohan Krishnamoorthy, Henry Dang**[**http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.646.8128&rep=rep1&type=pdf**](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.646.8128&rep=rep1&type=pdf)

Het process van het afnemen van temperatuur wordt een *cooling scheme* genoemd. Om cooling schemes te verbeteren zijn er een aantal technieken ontwikkeld die centraal staan rondom het idee om de neighborhood van een configuratie uit te breiden zodat er op deze manier de local optimum wordt ontweken. Voorbeelden hiervan zijn: the introduction of

a complicated neighbourhood structure, the use of sophisticated transition mechanisms and cooling schedules involving re-heating as well as cooling.

SA is een effectieve method om het CTP aan te pakken. Het algoritme gebruikt een veelgebruikte cooling scheme, namelijk ‘geometric cooling schedule’.

A cooling schedule consists of: (i) *Starting temperature* (ii) *Cooling factor* ; (iii) *Markov chain length*; and (iv) *Process termination condition*.

Er worden zes verschillende cooling schedules onderzocht:

1. **Geometric cooling schedule:** The cooling factor is assumed to be a constant and less than one. The simple geometric cooling scheme always applies the same cooling rate regardless of the state of the system. it is known that a substance may undergo a number of phase transitions during cooling, before it reaches its frozen state. At each phase transition the physical structure of the substance undergoes a change which alters its behaviour.
2. **Multiple cooling rates:** lijkt op de eerste, However, at high temperatures almost all proposed interchanges are (mostly) accepted, even though many of them could be non-productive. The application of a different cooling rate, that depends on the phase, would allow the algorithm to spend less time in the high temperature phases. Consequently, more time would be spent in the low temperature phases, thus reducing the total amount of time required to solve a particular problem .
3. **Geometric reheating:** no matter how slow the cooling rate, the algorithms will always deliver a local minima. The only solution to this problem is to detect that a local minima has been reached, and to reheat the system, this allowing it to escape from the local minima. In this scheme, it is necessary to determine when to swap between heating and cooling.
4. **Enhances geometric reheating:** This scheme is an enhanced version of the geometric reheating method. It is introduced to prevent excessive heating of the system and to overcome cycling, which occurs between heating and cooling. This is achieved by assigning a large heating value, which is close to 1 every time a local optimum is encountered. This value is then decreased if, after a fixed number of Markov chains, the system is still trapped in the local minima.
5. **Non-monotonic cooling schedule:**  A potential disadvantage of this approach is that it may take a number of cycles before the temperature is sufficiently high to allow significant changes to occur. In the non-monotonic cooling regime the temperature is reset in one step, thus removing the non productive cycles. Whenever the occurrence of a local minima is detected, this scheme attempts to escape from it by resetting the temperature to the maximum of either half the previous reset temperature or the temperature at which the best solution was attained.
6. **Reheating as a function of cost:** This directly sets the temperature to a certain value without the need to approach the desired temperature slowly. This scheme also allows the system to escape local minima. Thus, as the quality of solution improves, the system is reheated less.

🡪 In summary, the use of multiple cooling rates is more effective than one. The point at which cooling rates are switched is the temperature at which the specific heat is maximal to denote a phase change. We experimented with four schemes which raise the temperature in order to allow for the system to escape from local minima. In general, the scheme which uses the phase transition temperature in combination with the best solution quality found to date produced the best results