

Et lokalsøgningssystem til at løse diskrete optimeringsproblemer

Bo Stentebjerg-Hansen

Vejleder: Marco Chiarandini

Syddansk Universitet

Institut for Matematik og Datalogi

3. marts 2016

Slide 1

Overblik

- 1 Introduktion
- 2 Elementer i lokalsøgning
- 3 Opbygning af systemet
- 4 Lokalsøgningsalgoritmer
- 5 Experimentel evaluering

Slide 2

└ Introduktion

Diskrete optimeringsproblemer

En diskret optimeringsinstans I :

- n variable \mathbf{x} , $\mathbf{x} \in \mathbb{Z}^n$
- m betingelser \mathbf{C}
- Evaluerings funktion $f(\mathbf{x})$

$$\min\{f(\mathbf{x}) \mid \mathbf{x} \in \text{feas}(I)\}$$

Kan være NP-hårde problemer

Eksempel: Skemalægningsinstans

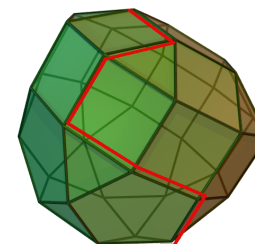
- Variable: Klasser der skal skemalægges
- Betingelser: Ingen overlap, krav til lokaler, tidspunkter, osv
- Evaluering: Totale antal af mellemtimer
- Bedste løsning: Den løsning som overholder alle betingelser og giver færrest mellemtimer

Slide 3

└ Introduktion

Løsningsmetoder

- Algoritmer til specifikke problemer
 - Approximations algoritmer
 - Dynamisk programmering
- Satisfiability solver
- Lineær heltalsprogrammering
- **Constraint Programming**
- **Lokalsøgning**
- Flere andre



Slide 4

Lineær helttalsprogrammering

- Model baseret på uligheder

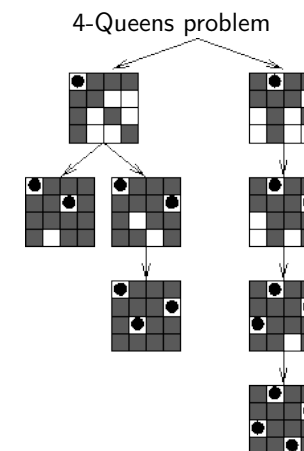
$$\begin{array}{ll} \text{Minimize} & z = 2x_1 + x_2 + x_3 \\ \text{subject to} & -x_1 + 2x_2 \leq 1 \\ & x_1 + x_2 + x_3 = 2 \\ & x_1, x_2, x_3 \in \mathbb{Z}^+ \end{array}$$

- Gurobi, CPLEX, SCIP, GLPK
- Kan ikke altid finde en (optimal) løsning inden for rimelig tid

Slide 5

Constraint Programming (CP)

- Bruger søgetræer + propagation til at finde en løsning
- Mere naturlig formulering af problemer
- Flere muligheder for at modellere et problem
- Men mindre egnet til optimeringsproblemer
- Fx Gecode, CHIP, Prolog



Slide 6

Lokalsøgning

- “Trial and error” teknik
- Alle variable skal have en værdi først
- Delt op implicite og bløde betingelser
- Ændre få variable ad gangen og beregner effekten.
- Kan ikke garantere optimalitet
- Ofte implementeret for fra til specifikke problemer.

2	6	3	7	8	9	4	1	5			6	8	4	7	3	5	1	2	9	
5	8	9	4	1	3	7	6	2			5	1	2	4	6	9	3	7	8	
1	4	7	2	5	6	8	9	3			7	9	3	2	1	8	4	5	6	
7	9	8	6	2	1	5	3	4			4	2	5	9	7	6	8	1	3	
4	5	1	8	3	7	9	2	6			8	3	1	5	2	4	9	6	7	
6	3	2	5	9	4	1	8	7			9	7	6	1	8	3	2	4	5	
3	1	5	9	4	2	6	7	8	2	3	4	1	5	9	3	4	7	6	8	2
9	7	4	3	6	8	2	5	1	6	7	9	3	4	8	6	5	2	7	9	1
8	2	6	1	7	5	3	4	9	5	8	1	2	6	7	8	9	1	5	3	4

Slide 7

Sammensætning af løsningsmetoder

- LocalSolver¹
 - Matematisk modellering
 - Lineær- og helttalsprogrammering
 - Kommerciel Solver
- EasyLocal++²
 - Lokalsøgningsalgoritmer implementeret
 - Bruger definere hvilke der skal bruges
- Comet³
 - Først betingelsesbaseret lokalsøgningsystem
 - Mulighed for udvidelse
 - Ikke længere vedligeholdt
- Oscar⁴
 - Inspireret af Comet
 - Forholdsvis nyt

¹<http://www.localsolver.com/>

²<https://bitbucket.org/satt/easylocal-3>

³Constrained-Based Local Search, P. Hentenryck & L. Michel

⁴<http://oscarlib.bitbucket.org/cbls.html>

Slide 8

Hvorfor et andet lokalsøgningssystem?

- Kombinerer CP og lokalsøgning på en ny måde:
 - Ikke kun modelleringsmæssigt
 - Ikke som “large scale neighborhood search”
- Undersøger effekten af (offline) CP domæne reducering
- Bruger CP som konstruktions heuristik
- Tester effekten af envejsbetingelser
- Introducerer en ny evalueringsmetode
 - Leksikografisk vægtning

Slide 9

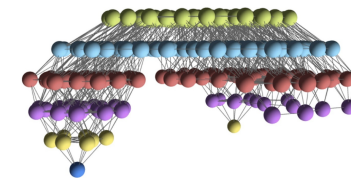
Begreber i lokalsøgning

Modellering:

- Variable:
 - løsningsrepræsentation, search space $S(I)$
- Betingelser:
 - Bløde betingelser
 - Envejsbetingelser
 - Implicitte betingelser
- Evaluerings funktion: kvaliteten af en løsning

Søgning:

- Neighborhood
- Konstruktions heuristik
- Lokalsøgning
- Metaheuristikker
 - Tabu søgning
 - Iterativ lokalsøgning
 - ...



Slide 10

Binære optimeringsproblemer

$$\min \left\{ \sum_{j=1}^n c_j x_j \mid \sum_{j=1}^n a_{ij} x_j \leq b_j, \forall i \in \{1..m\} \right\}$$

Minimize $z = 2x_1 + x_2 + x_3$
 subject to $-x_1 + 2x_2 \leq 1$
 $x_1 + x_2 + x_3 = 2$
 $x_1, x_2, x_3 \in \{0, 1\}$

- Mange problemer kan modelleres som binære optimeringsproblemer
- Fx: traveling salesman problem, knapsack, vertex cover, ...

Bliver oprettet i dette system med **linear**:

linear(*int*[] coefficients, *Variable*[] variables, *int* relation, *int* bound, *int* priority)

Slide 11

Før lokalsøgning kan startes

Find en startløsning:

- Opret variable og begrænsninger
- Domæne reducering lavet af Gecode
- Oprettelse af søgningsstrategi til Gecode
- Find en gyldig løsning
 - til 50 % af betingelserne
 - til 25 % af betingelserne
 - til 12,5 % af betingelserne
 - Tilfældig tildeling af værdi til variable inden for deres domæne

Behandling af betingelser:

- 1 Invarianter: Variable defineret af betingelser
- 2 Graf over afhængighed mellem variable og invarianter
- 3 Auxiliary invarianter: Betingelser behandlet som invarianter
- 4 Topologisk sortering af invarianter.

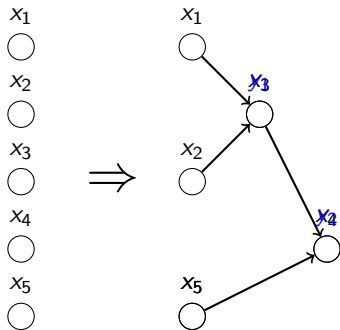
Slide 12

Invarianter og afhængighedsgraf

- Betingelses bestemt om envejsbetingelse kan laves

$$x_1 + x_2 - x_3 = 1 \Leftrightarrow x_3 = x_1 + x_2 - 1$$

$$x_3 - x_4 + x_5 = 1 \Leftrightarrow x_4 = x_3 + x_5 - 1$$

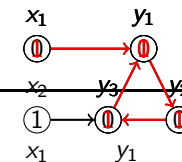


- Færre mulige løsninger der skal undersøges
- Bruger lidt mere tid på at evaluere en løsning
- x_3 er gjort afhængig af x_1 og x_2
- x_4 indirekte afhængig af x_1 og x_2
- Variable valgt efter antal udgående kanter og antallet af betingelser den optræder i

Slide 13

Kredse i grafen

$$\begin{aligned} y_1 &= x_1 - y_3 \\ y_2 &= y_1 \\ y_3 &= x_2 + y_2 - 1 \\ x_1, x_2 &\in \{0, 1\} \\ y_1, y_2, y_3 &\in \{0, 1\} \\ y_1 &= x_1 - y_3 \\ y_2 &= y_1 \\ y_3 &= x_2 + y_2 - 1 \\ x_1, x_2 &\in \{0, 1\} \\ y_1, y_2, y_3 &\in \{0, 1\} \\ y_1 &= x_1 - y_3 \\ x_3 - y_1 &= 0 \\ y_3 &= x_2 + x_3 - 1 \\ x_1, x_2, x_3 &\in \{0, 1\} \\ y_1, y_3 &\in \{0, 1\} \end{aligned}$$



Identificering af kredse:

- Dybde først lignende algoritme, af Tarjan $O(V + E)$
 - Finder stærke sammenhængskomponenter (SCC)
 - Fjerne en invariant \rightarrow genskaber en variable fra hver SCC
 - Vælger invariant efter antal indgående kanter
- Gentages indtil ingen stærke sammenhængskomponenter er fundet
- Balance mellem tid og effekt

Slide 14

Yderligere invarianter

- Betingelser som ikke er brugt til at definere variable
- Betingelses specifik oprettelse af invarianter
- Tilføj invarianter til grafen
- Invarianter til summering af overtrædelse betingelser

For en **linear** betingelsen:

- Summering af venstresiden:

$$\underbrace{x_1 + 2x_2 - x_3}_{w_1} \leq 2$$

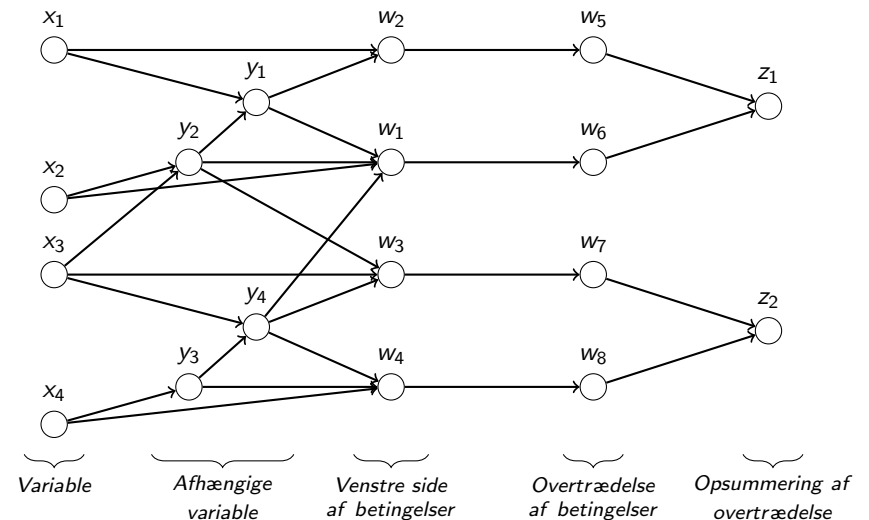
- Overtrædelse af betingelsen:

$$\underbrace{w_1}_{w_2} \leq 2$$

$$w_2 = \begin{cases} w_1 - 2, & \text{if } w_1 > 2. \\ 0, & \text{otherwise.} \end{cases}$$

Slide 15

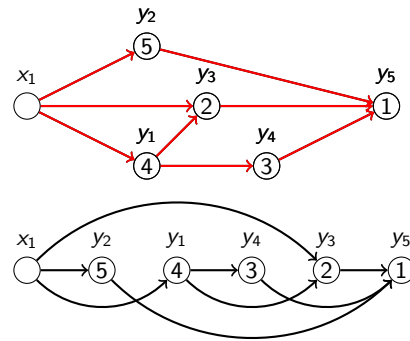
Endelige graf



Slide 16

Ordning af invarianter

- Lav ordning af invarianter til når de skal opdateres
- Forhindre flere opdateringer af samme invariant
- Ordningen kan laves med dybde først søgning i grafen
- Opret en liste for hver uafhængig variable



Slide 17

Basis for algoritmerne

Billed af busstop

Optimal løsning

FI

BI

RW

Slide 18

Test og resultater

Slide 19