Et Lokalsøgningssystem til at Løse Diskrete Optimeringsproblemer

Bo Stentebjerg-Hansen

Syddansk Universitet

Institut for Matematik og Datalogi

12. februar 2016

Overview

Introduktion

-

Introduktion

-
- $O(log^* n + \Delta)$ rounds, Δ is the highest vertex degree

Binære optimeringsproblemer

$$\text{Minimize} \quad z = \mathbf{c}^T\mathbf{x} \tag{1}$$
$$\text{subject to} \quad \mathbf{Ax} \leq \mathbf{b} \tag{2}$$
$$\mathbf{x} \in \{0, 1\}^n \tag{3}$$

$\mathbf{A}$ er en $m \times n$ matrice, $\mathbf{c}$ og $\mathbf{b}$ er n dimensionale vectorer, alle tre består af heltal. $\mathbf{x}$ er en n dimensional vector bestående af binære variable.

Eksempel

$$\begin{array}{llrcrcrl}
\text{Minimize} & z = & 2x_1 & + & x_2 & + & x_3 & \\
\text{subject to} & & -x_1 & + & 2x_2 & & & \leq 1 \\
 & & x_1 & + & x_2 & + & x_3 & = 2 \\
 & & \multicolumn{5}{l}{x_1, x_2, x_3 \in \{0, 1\}} &
\end{array}$$

En mulig løsning:

$x_1 = 1$
$x_2 = 0$
$x_3 = 1$

$$z = 2 \cdot 1 + 1 \cdot 0 + 1 \cdot 1 = 3$$

Helttals programmering

- Simplex metode.

Helttals programmering

- Simplex metode.
- Ligningsbaseret model.

Helttals programmering

- Simplex metode.
- Ligningsbaseret model.
- Gurobi, GLPK, SCIP.

Constraint programming

- Bruger søgetræer til at finde en løsning.

Constraint programming

- Bruger søgetræer til at finde en løsning.
- Mere naturlig formulering af problemer.

Constraint programming

- Bruger søgetræer til at finde en løsning.
- Mere naturlig formulering af problemer.
- bl.a. Gecode, prolog.

Lokal søgning

- Undersøger mange små ændringer.

Lokal søgning

- Undersøger mange små ændringer.
- Kan ikke garantere optimalitet.

Lokal søgning

- Undersøger mange små ændringer.
- Kan ikke garentere optimalitet.
- Ofte skrevet til et specifikt problem.

Constraint programming med lokal søgning

- Formulering af problem som i Constraint programming.

Constraint programming med lokal søgning

- Formulering af problem som i Constraint programming.
- Genanvendelse af algorithmer.

Constraint programming med lokal søgning

- Formulering af problem som i Constraint programming.
- Genanvendelse af algorithmer.
- Giver mulighed for at fokusere på modellering.

Constraint programming med lokal søgning

- Formulering af problem som i Constraint programming.
- Genanvendelse af algorithmer.
- Giver mulighed for at fokusere på modellering.
- Solver fx Comet og OscaR.

Det er projekt

- Kombinere Gecode og lokal søgning.

Det er projekt

- Kombinere Gecode og lokal søgning.
- Undersøger effekten af Gecode.

Det er projekt

- Kombinere Gecode og lokal søgning.
- Undersøger effekten af Gecode.
- Tester brugen af invarianter.

Det er projekt

- Kombinere Gecode og lokal søgning.
- Undersøger effekten af Gecode.
- Tester brugen af invarianter.
- Introducere en ny evaluerings metode.

Overblik

- Objekter, en kasse med værktøj og information.

Her er et flot billed af formulering -¿ GPSolver -¿ GecodeEngine -¿ LocalSearcEngine.

Overblik

- Objekter, en kasse med værktøj og information.
- Brugerflade og to delt system.

Her er et flot billed af formulering -¿ GPSolver -¿ GecodeEngine -¿ LocalSearcEngine.

Brugen af Gecode

Brugen af Gecode

- Opret variable og begrænsninger.

Brugen af Gecode

- Opret variable og begrænsninger.
- Preprocessering af Gecode.

Brugen af Gecode

- Opret variable og begrænsninger.
- Preprocessering af Gecode.
- Oprettelse af søgningsstrategi.

Brugen af Gecode

- Opret variable og begrænsninger.
- Preprocessering af Gecode.
- Oprettelse af søgningsstrategi.
- Finder måske en gyldig løsning.

Transformation af Modellen

-

Transformation af Modellen

Transformation af Modellen

-
-
-

Transformation af Modellen

Maximal matching for G

The maximal matching of $\mathcal{M} = M_1, \ldots, M_\Delta$ can be computed. The maximal matching M_1 of forest F_1 is computed and the vertices removed from V. Then the maximal matching M_2 is computed and vertices removed and so on for all forests $F'_3, \ldots, F'_\Delta$. This gives a total of 3Δ rounds

Algorithm

Algorithm 1 pseudocode for maximal matching

Compute forest decomposition $F_1 \dots F_\Delta$
Make all edges directed from lowest ID to higher ID
Compute 3-coloring of each forest F_i in parallel
$\mathcal{M} \leftarrow \emptyset$
$V' \leftarrow V$
for $i \leftarrow 1$ to Δ **do**
    for $j \leftarrow 1$ to 3 **do**
        let c_j be the set of vertices colored j in F_i
        Every $u \in c_j \cap V'$ selects one outgoing edge in (V', E_i)
        Let M_j be the set of edges selected
        $\mathcal{M} \leftarrow \mathcal{M} \cup M_j$
        $V' \leftarrow V' \setminus \{u \mid u \text{ is matched in } M_j\}$
    end for
end for

Analysis

- First the graph G is decomposed into at most Δ forests in constant time
- In each forest the trees can be 3-colored in $O(log^* n)$ rounds in parallel
- Computing the maximal matching in a tree takes three rounds
- There are Δ forest hence the number of rounds for the matching is $O(\Delta)$
- The total number of rounds need is $O(log^* n + \Delta)$

Questions?

Questions?

The End

Thanks for your attention