FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

A Hybrid Algorithm for the Partition Coloring Problem

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Computational Intelligence

eingereicht von

Gilbert Fritz
Matrikelnummer 0827276

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Univ.-Prof. Dipl.-Ing. Dr.techn. Günther Raidl
Mitwirkung: Univ.Ass. Dipl.-Ing. Dr.techn. Dr. Bin Hu

Wien, 21.Oct.2013

(Unterschrift Verfasserin)	(Unterschrift Betreuung)

Erklärung zur Verfassung der Arbeit

Gilbert Fritz
Schlosshofer Straße 49/18

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum) (Unterschrift Verfasserin)

Danksagung

Ich danke meinen Betreuern, ao. Univ.-Prof. Dipl.-Ing. Dr.techn. Günther Raidl und Univ.Ass. Dipl.-Ing. Dr.techn. Bin Hu für ihre Unterstützung bei der Erstellung dieser Arbeit durch ihr konstruktives Feedback und ihre Ideen, welche es mir ermöglicht haben, immer neue Aspekte der Problemstellung zu erkennen.

Besonderer Dank gilt meinen Eltern, Franz und Justine Fritz, sowie meiner Partnerin Odnoo und meinen Freunden für Ihre Unterstützung.

Abstract

The Partition Coloring Problem (PCP) is a generalization of the the classical Vertex Coloring Problem (VCP), partitioning the set of nodes into clusters and seeking a coloring for the subgraph induced by selecting exactly one node from each cluster.

Diese Arbeit beschäftigt sich mit dem Partition Coloring Problem (PCP). Es handelt sich dabei um eine Generalisierung des Knotenfärbungsproblems und ist ein Optimierungsproblem der Komplexitätsklasse $\mathcal{NP}$.
Gegeben ist ein Graph, dessen Knotenmenge in disjunkte Partitionen unterteilt ist. Aus jeder Partition muss ein Knoten gewählt werden. Der durch die gewählten Knoten induzierte Subgraph soll unter der Bedingung eingefärbt werden, dass kein zueinander adjazentes Knotenpaar die gleiche Farbe annimmt. Ziel ist es, die Gesamtanzahl der verwendeten Farben - die sogenannte chromatische Zahl - zu minimieren.
Zur Lösung dieses Problems sollen mittels heuristischer Verfahren initiale Lösungen erstellt und diese mittels Tabusuche und wiederholter, partieller Neueinfärbung verbessert werden. Das Problem der Neueinfärbung wird mit unterschiedlichen Ansätzen gelöst.

Kurzfassung

Todo

Contents

CHAPTER 1

Problem Definition

CHAPTER 2

Problem Solving Approach

In this chapter, the algorithms and models of the new hybrid approach for the PCP will be described and analysed in detail. First in section 2.1, the main procedure is explained. Section 2.2 then analyses the two different construction heuristics used in this work, namely *OneStepCD* and *DANGER*. The improvement phase is split into two parts: algorithms, that assign a new, but not mandatorily feasible coloring to a chosen set of nodes, here consisting of one random-, one heuristic- and two exact approaches, are applied in the first part. Below, this thesis will refer to these algorithms to as the "recoloring" algorithms 2.3. In section 2.4, the tabu search, which tries to find a coloring that makes the conflict-prone solution created by one of the recoloring algorithms feasible is presented. Finally, two variants, one of them consisting in a modified ILP formulation and the other one in adding lately recolored areas to the tabulist, are considered in section 2.5.

2.1 Main Procedure

The idea of the approach in general is – starting from a feasible solution – to pick a color and eliminate it. For each color c, the set of nodes colored with c is reselected as well as recolored without using color c. Using this strategy, it might not be possible to find a feasible solution. Therefore, infeasible solutions are accepted, i.e. solutions including at least one conflict, that is, a pair of adjacent nodes $(i, j) \in E, i \in V, j \in V$ colored with the same color. One of the two nodes is chosen and in the following refered to as the "conflicting node"[1]. These conflicting nodes form the starting points for the tabu-search, that tries to find an alternative color for each conflicting node. This process eliminates the conflicting node, but eventually produces further conflicts, which then again have to be eliminated. If no feasible coloring can be found withing a specified number of iterations, the next color $c + 1$ is considered. If all conflicts can be eliminated, the algorithm has successfully decreased the chromatic number and repeats the whole process.

[1] How the node is chosen is shown in detail in 2.3

This approach differs from the strategy presented in [24] mainly by the effort that is investigated in the recoloring phase. There, Noronha et.al. reassign colors in a random way and therefore produce a random number of conflicts. The main innovation of the idea presented in this work is the minimization of the number of conflicts produced by an advanced recoloring algorithm, in order to increase the chance of eliminating these conflicts by the tabu search.

TODO: INSERT GRAPHIC!!!!

Listing 1 provides an overview of the single steps that have been implemented. The algorithm takes an instance P of PCP, an algorithm $INITIAL$ computing an initial solution, and a recoloring algorithm $RECOLOR$ as input. As described in 1, an instance of PCP consists of an uncolored graph $G = (V, E)$, where V is divided into k clusters. Parameter $INITIAL$ can be any algorithm that creates a feasible solution for PCP. Two of them have been taken into account in this work and are described in section 2.3, others are proposed e.g. in [21].
In line 1, the initial solution is calculated and assigned to S. The chromatic number of S is assigned to $cmax$ in line 2. Line 3 initializes an empty set X. Line 5 to line 9 are preformed for each color $c \in \{1, \ldots, cmax\}$. In line 5 all nodes in V are selected that are colored with color c and denoted by V_c. In line 6 a copy S' of S is created and there all nodes in V_c are recolored by the algorithm $RECOLOR$ excluding color c. The set of conflicting nodes is denoted by C_c.

Algorithm 1: PCP Hybrid

Input: A problem instance $\mathcal{P}$, an algorithm $INITIAL$ and an algorithm $RECOLOR$
Output: A feasible Solution S

$S \leftarrow INITIAL(\mathcal{P})$;
$cmax \leftarrow$ the chromatic number of S;
Set $X \leftarrow \emptyset$;
for $c = 1, \ldots, cmax$ **do**
    Let V_c be the set of nodes coloured by the colour c;
    Let S_c be the solution created by applying $RECOLOR(V_c, \{1, \ldots, cmax\} \setminus c)$ on S;
    Let C_c be the set of all nodes involved in color conflicts of S_c;
    $X \leftarrow X \cup \langle S_c, V_c, C_c \rangle$
Sort elements X ascendingly by $|C_i|$;
$reduction \leftarrow$ false;
for $(S_c, V_c, C_c) \in X$ **do**
    $S'_c \leftarrow TabuSearch(S_c, V_c, C_c)$;
    if S'_c *is free of conflicts* **then**
        $reduction \leftarrow$ true;
        break;
if $reduction$ **then**
    $S \leftarrow S_c$;
    $cmax = cmax - 1$;
    goto line 3;
return S;

2.2 Constructional Heuristics

Algorithm 2: OneStepCD

Input: An uncolored Graph $G = (V, E)$

Output: A feasible Coloring V'

```
Remove from G all edges (i, j) ∈ E : i, j ∈ V_k for some k = 1, ..., q;
Set V' ← ∅;
while |V'| < q do
    Set X ← ∅;
    for k = 1, ..., q : V_k ∩ V' = 0 do
        Set X ← X ∪ argmin{CD(i) : i ∈ V_k};
    Set x ← argmax{CD(i) : i ∈ X};
    Set V' ← V' ∪ {x};
    Assign the minimum possible colour to x;
    Remove from G all nodes in V_c(x) \ {x};
return V';
```

2.3 Recoloring

Random

OneStepCD

Algorithm 3: OneStepCD Recoloring

```
Input: A partial Solution P, a number of maximum colours cmax
Output: A feasible Solution S
Let U be the set of uncolored nodes in P;
Set S ← ∅;
while |U| > 0 do
    Set X ← ∅;
    for u ∈ U do
        X ← X ∪ argmin{CD(i) : i ∈ V_{c(u)}};
    Set x ← argmax{CD(i) : i ∈ X};
    Set cmin ← the minimum possible colour that can be assigned to x;
    if cmin ≥ cmax then
        cmin ← the color that produces the fewest conflicts.
    Assign cmin to x;
    S ← S ∪ {x};
    U ← U \ V_{c(x)};
return V';
```

ILP minimizing conflicts

Let $Q = Q_1, \ldots, Q_q$ be the set of Clusters. Every cluster Q_p consists of a set of nodes. Let $C = \{1, \ldots, cmax\}$ be the set of allowed colors. Let M be a 3-dimensional array of constants, storing for every cluster $p \in Q$, the number conflicts that would occur by selecting the pair $(v \in Q_p, c \in C)$. E denotes the set of edges and $P[v]$ the cluster of node v.

$$
[h]
\begin{aligned}
\underset{X}{\text{minimize}} \quad & \sum_{p \in Q} \sum_{v \in Q_p} \sum_{c \in C} X_{pvc} * M_{pvc} && && (1) \\
\text{subject to} \quad & \sum_{v \in Q_p} \sum_{c \in C} X_{pvc} = 1, && \forall p \in Q && (2) \\
& X_{pvc} + X_{quc} \leq 1, && \forall ((p,v),(q,u)) \in E, \forall c \in C && (3) \\
& X_{pvc} \in \{0, 1\}, && \forall p \in Q, \forall v \in Q_p, \forall c \in C && (4)
\end{aligned}
$$

ILP minimizing conflicting nodes

Let U be the set of uncolored nodes in uncolored clusters and $color[(p,v)]$ the color of the node v in partition p.

$$
[h]
\begin{aligned}
\underset{Z}{\text{minimize}} \quad & \sum_{p \in Q} \sum_{v \in Q_p} \sum_{c \in C} Z_{pvc} && && (1) \\
\text{subject to} \quad & Z_{pvc} \geq X_{quc}, && \forall ((p,v),(q,u)) \in E : (p,v) \notin U, (q,u) \in U, c = color[(p,v)] && (2) \\
& \sum_{v \in Q_p} \sum_{c \in C} X_{pvc} = 1, && \forall p \in Q && (3) \\
& X_{pvc} + X_{quc} \leq 1, && \forall ((p,v),(q,u)) \in E, \forall c \in C && (4) \\
& X_{pvc} \in \{0, 1\}, && \forall p \in Q, \forall v \in Q_p, \forall c \in C && (5)
\end{aligned}
$$

2.4 Tabu Search

2.5 Variants

Algorithm 4: TabuSearch

Input: An infeasible solution S, the set of previously recolored nodes R, the set of conflicting nodes C

Output: A Solution $\bar{S}$

Set $C \leftarrow C \setminus R$;
Set $cmax \leftarrow$ the chromatic number of S;
Set $iter \leftarrow 0$;
Set $minConflicts \leftarrow \infty$;
Set $\bar{S} \leftarrow S$;
while $|C| > 0$ *and* $iter < maxiter$ **do**
    for $V_{c(u)} : u \in C$ **do**
        for $v \in V_{c(u)}$ *and* ***for*** $c = 1, \ldots, cmax$ **do**
            Obtain a tentative solution S' by selecting and coloring node v with color c in $\bar{S}$;
            if $conflicts(S') = 0$ **then**
                $\bar{S} \leftarrow S', \bar{v} \leftarrow v, \bar{c} \leftarrow c$;
                goto line 16;
            else if *the pair* v, c *is not in the tabu list* **then**
                if $conflicts(S') < minConflicts$ **then**
                    $minConflicts \leftarrow conflicts(S')$;
                    $\bar{S} \leftarrow S', \bar{v} \leftarrow v, \bar{c} \leftarrow c$;
    insert pair $\bar{v}$, $\bar{c}$ in the tabu list for TabuTenure iterations;
    $C \leftarrow C \setminus u$;
    Let $C_{\bar{v}}$ be the set of nodes conflicting with $\bar{v}$;
    $C \leftarrow C \cup C_{\bar{v}}$;
return $\bar{S}$;

Bibliography

[1] R. Andersen. Finding large and small dense subgraphs. *CoRR*, 0707032, 2007.

[2] D. Applegate, R. Bixby, V. Chvátal, and W. Cook. Solution of a 15112-city traveling salesman problem. *http://www.math.uwaterloo.ca/tsp/d15sol/index.html*, 2001.

[3] M. Templ B. Meindl. Analysis of commercial and free and open source solvers for linear optimization problems. 2012.

[4] Lucile Belgacem, Irène Charon, and Olivier Hudry. A post-optimization method for the routing and wavelength assignment problem applied to scheduled lightpath demands. *European Journal of Operational Research*, 2013.

[5] C. Blum and A. Roli. *Metaheuristics in combinatorial optimization: Overview and conceptual comparision*. ACM Computing Surveys, 2003.

[6] M.Gurusamy C. S. Ram Murthy. Wdm optical networks - concepts. *Design and Algorithms*, 2002.

[7] M. Charikar. Greedy approximation algorithms for finding dense components in a graph. *APPROX*, (84-95), 2000.

[8] Raymond Chiong. Nature-inspired algorithms for optimisation. *Studies in Computational Intelligence*, 193, 2009.

[9] C.Volko. Selective graph coloring problem. 2012.

[10] Bioinspired Computation F. Neumann, C. Witt. Bioinspired computation in combinatorial optimization. *Combinatorial Optimization, Natural Computing Series*, 2010.

[11] C. Feremans. Generalized network design problems. *European Journal of Operational Research*, 148(1-3), 2003.

[12] Yuri Frota, Nelson Maculan, Thiago F. Noronha, and Celso C. Ribeiro. A branch-and-cut algorithm for the partition coloring problem. *Networks*, 55(3):194–204, 2010.

[13] M. R. Garey and D. S. Johnson. Computers and intractability: A guide to the theory of np-completeness. *Series of Books in the Mathematical Sciences*, 1, 1979.

[14] M. Gendreau. *Handbook of metaheuristics*. Springer, 2010.

[15] S.Pirkwieser G.Fritz, G.Raidl. Heuristic methods for the hop constrained survivable network design problem. 2011.

[16] F. Glover. Future paths for integer programming and links to artificial intelligence. *Cambridge University Press*, 13(533–549), 1986.

[17] S. Wright J. Nocedal. Numerical optimization. 2000.

[18] R. M. Krishnaswamy and K. N. Sivarajan. Algorithms for routing and wavelength assignment based on solutions of lp-relaxations. *IEEE Communication Letters*, 5(10), 2001.

[19] E. Lawler. Combinatorial optimization: Networks and matroids. 2002.

[20] E.L. Lawler. Combinatorial optimization: Networks and matroids. 1976.

[21] Guangzhi Li and Rahul Simha. The partition coloring problem and its application to wavelength routing and assignment. In *1st Workshop on Optical Networks*, 2000.

[22] Goran Z. Marković and Vladanka S. Aćimović-Raspopović. Generalized network design problems. *Telfor Journal*, 2010.

[23] B.Hu M.Leitner, G.Raidl. Solving two generalized network design problems with exact and heuristic methods. 2006.

[24] Thiago F. Noronha and Celso C. Ribeiro. Routing and wavelength assignment by partition colouring. *European Journal of Operational Research*, 171(3):797–810, 2006.

[25] I. H. Osman and G. Laporte. Metaheuristics: A bibliography. *Annals of Operations Research*, 63(513-623), 1996.

[26] H.Romeijn P. Pardalos. Handbook of global optimization. 2, 2002.

[27] C. M. Papadimitriou. Computational complexity. 1, 1994.

[28] S.Pirkwieser P.Gebhard, G.Raidl. The vehicle routing problem with compartments based on solutions of lp-relaxations. 2012.

[29] Petrica C. Pop, Bin Hu, and Günther R. Raidl. A memetic algorithm for the partition graph coloring problem. In *Extended Abstracts of the 14th International Conference on Computer Aided Systems Theory*, pages 167–169, Gran Canaria, Spain, 2013.

[30] B. Saha S. Khuller. On finding dense subgraphs. 2009.

[31] K. Jansen T. Erlebach. The complexity of path coloring and call scheduling. *Theoretical Computer Science*, 255, 2001.

[32] G. Kortsarz U. Feige and D. Peleg. The dense k-subgraph problem. *Algorithmica*, 29(410-421), 1997.

[33] Ingo Wegener. *Complexity Theory: Exploring the Limits of Efficient Algorithms*. Springer, 2005. Reflects recent developments in its emphasis on randomized and approximation algorithms and communication models.

[34] David P. Williamson and David B. Shmoys. The design of approximation algorithms. *Cambridge University Press*, 2010.

[35] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(67-82), 1997.

[36] H. Tamakic T. Tokuyamad Y. Asahiroa, K. Iwamab. Greedily finding a dense subgraph. *Journal of Algorithms*, 34, 2000.

[37] K.Iwama Y.Asahiro, R.Hassin. Complexity of ÿnding dense subgraphs. *Discrete Applied Mathematics*, 121(15–26), 2002.