A „Quotient and Power methods for the Graph Colouring Problem"

című doktori értekezés összefoglalója

Szerző: *István Juhos*

Ph.D. candidate

Department of Computer Algorithms and Artificial Intelligence

Témavezető: *Prof. János Csirik*

Head of the
Department of Computer Algorithms and Artificial Intelligence

Szeged 2009

University of Szeged
Ph.D. School in Computer Science

Bevezetés

Ezen összefoglaló tartalmazza a szerző „Quotient and Power methods for the Graph Colouring Problem„ című doktori disszertációjának eredményeit. A szerző kifejlesztett egy általános keretrendszert gráfszínezési algoritmusok számára, ahol a hagyományos színezés speciális gráf homomorfizmusokon keresztül került definiálásra Zykov munkássága nyomán [64; 65]. Ezen homomorfizmusok hasznosnak bizonyultak az algoritmus tervezésben (lásd Juhos et al. ([35; 37–43]). Ezen összefoglaló az értekezés struktúráját követi, amelyben Az eredmények a kifejlesztett gráfszínezési keretrendszernek megfelelően különböző részekre tagolódnak.

A szerző a gráfszínezési folyamatot kvóciens és hatványgráfok segítségével, gráf homomorfizmusokon sorozatával definiálta (lásd Juhos et al. [37; 41]). A szerző Kvóciens és Hatvány Módszernek nevezte el az ezen elven alapuló színezési módszereit. Továbbá ezen gráf homomorfizmusokat mátrixok és azokon értelmezett megfelelő műveletek segítségével határozta meg, melyet a színezési folyamat *Merge Modelljeinek* nevezett el [37; 40; 41]. A tárgyalt mátrix műveletek, *Merge Műveletek* egymásutánja hoz létre egy hagyományos értelemben vett színezést. A szerző a Merge Modelleken nyugvó stratégiákat, *Merge Stratégiákat* dolgozott ki [35; 38; 42; 43] amelyek lehetséges irányokat határoznak meg egy megoldás felé. Továbbá a szerző megadott három általános keretrendszert (*Merge Keretrendszert*) amelyekbe ezen stratégiák beágyazhatók [38; 40]. A keretrendszerek általánosítják a hagyományos szekvenciális színezési sémákat, így a már létező algoritmusok ezekbe beágyazhatók, amely azok általánosításához vezet. Egy ilyen beágyazás jelentős futási és színezési teljesítménynövekedést eredményezhet. Továbbá a keretrendszerek a stratégiákkal karöltve alkotnak konkrét algoritmusokat, *Merge Algoritmusokat*. A szerző több algoritmust is elkészített, amelyek a kísérleti vizsgálatok során felülmúltak több standard 'benchmark' algoritmust standard 'benchmark' gráfokon és az ún. 'phase transition' területen generált 'nehéz' gráfokon is [35; 37–43]. Egy-egy ilyen algoritmus az adott Merge Modell Merge Műveleteinek sorozatát generálja az alkalmazott Merge Stratégia vagy Stratégiák szerint, amely az eredeti gráfszínezési probléma egy lehetséges megoldását szolgáltatja.

A gráfszínezési probléma

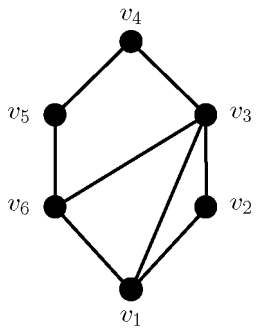
A gráf két véges diszjunkt halmazból álló halmazpár $G = (V, E)$, ahol $E \subseteq V \times V$. A V halmaz elemei a G gráf csúcsai, míg az E halmaz elemei alkotják G éleit. *A gráf csúcs k-színezés* (vagy röviden gráf k-színezés) a gráf csúcsaihoz színeket rendel egy k elemű színhalmazból, C-ből. A probléma akkor jelentkezik amikor a színezésnél figyelembe kell vennünk az éleket.

Definition 1 (Helyes gráf k-színezés) *A gráf $G = (V, E)$ egy helyes k-színezése, ha létezik, egy olyan k-színezés, ahol a szomszédos csúcsok különböző színeket kapnak:*

$$c : V \xrightarrow{sur} C \quad , \quad v_i \mapsto c(v_i) \quad , \quad \forall (v_i, v_j) \in E \Rightarrow c(v_i) \neq c(v_j) \quad , \quad |C| = k$$

Definition 2 (Gráf minimális csúcsszínezése) *A gráf $G = (V, E)$ egy minimális csúcsszínezése egy olyan helyes gráf χ-színezés, ahol χ a legkevesebb szín amely egy lehetséges k-színezésben elérhető.*

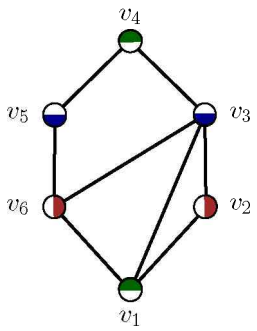
A χ-t a G gráf **kromatikus számának** nevezzük. A következőkben a gráfszínezési probléma a gráf minimum csúcsszínezési problémáját jelenti. Egy helyes színezésre példa az 1. ábrán látható. Az 1(a). ábra egy gráfot ábrázol $v_i \in V, i \in \{1, 2, \ldots, 6\}$ csúcsokkal/csomópontokkal és a köztük



(a) Egy gráf

	v_1	v_2	v_3	v_4	v_5	v_6
v_1	0	1	1	0	·	1
v_2	1	·	1	·	·	·
v_3	1	1	·	1	·	1
v_4	0	·	1	0	1	·
v_5	·	·	·	1	·	1
v_6	1	·	1	·	1	·

(b) Szomszédsági mátrix



(c) Egy helyes színezés

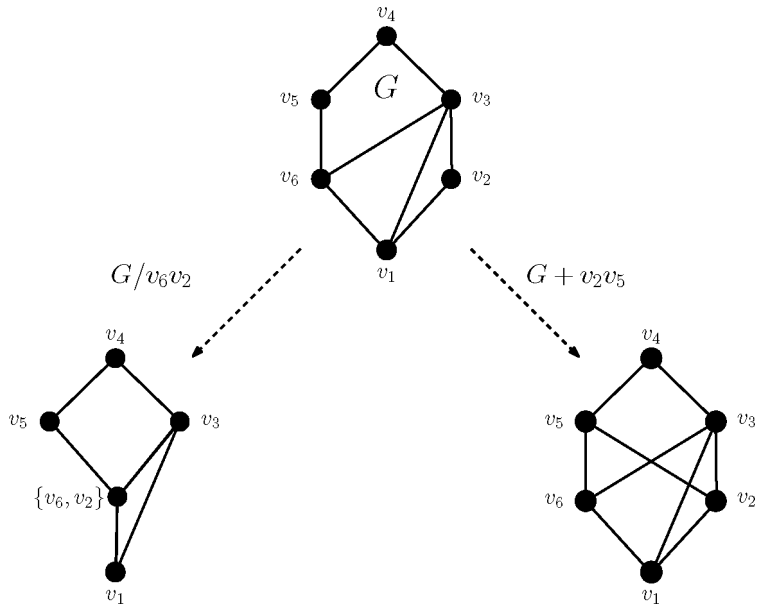
1. ábra. Egy gráf és annak egy helyes 3-színezése, amely egyben minimális színezés is.

futó élekkel. Mellette az 1(b). ábrán a gráf szomszédsági mátrixa található, amely az élek által definiált relációt határozza meg[1], továbbá a 1(c). ábra a gráf egy minimum színezését mutatja, ahol $\{v_1, v_4\}, \{v_2, v_6\}$ és $\{v_3, v_5\}$ csúcshalmazok alkotják a színosztályokat. A színosztályoknak független csúcshalmazoknak[2] kell lenniük, ahhoz hogy helyes színezést kapjunk. A 0-ák a 1(b). ábrán egy független csúcshalmazt jelölnek, a $\{v_1, v_4\}$ színosztályt.

Számos algoritmus ismert az irodalomban a minimum színezési probléma megoldására. Ezen algoritmusok két osztályba sorolhatók: az egzakt algoritmusok, ahol az optimális megoldás garantált, azonban a megoldásra fordított idő tekintélyes lehet a probléma komplexitásának köszönhetően (- amely NP-teljes [45]); és a nem egzakt algoritmusok, azaz az approximációs algoritmusok, ahol az optimális megoldás nem garantált, de sok esetben jó közelítő megoldást képesek adni belátható időn belül. Az utóbbi tartalmazhat sztochasztikus elemet is. Átfogó összefoglaló a következő publikációkban található ezekről a módszerekről: [23; 32; 47; 63]. A gráfszínezési probléma megoldható egzakt módon a keresési tér egy szisztematikus bejárásával [15; 16; 34]. Sajnos ez a módszer hatékonyan nem alkalmazható, mert már kisebb gráfok is hatalmas keresési teret eredményezhetnek, így a futási ideje ennek a keresési megoldásnak nagyon megnő. Ezen módszer hatékonyságának növelése érdekében számos heurisztikát fejlesztettek ki, hogy egy jó kezdeti lehetséges megoldást generáljanak, amely közel van a kívánt optimumhoz [4; 17; 26–28; 46; 49; 55; 58; 60–62]. Majd a generált lehetséges megoldásból kiindulva egy szisztematikus keresés jelentősen javíthatja a kezdeti megoldást.

Általában a keresés a generált megoldás környezetét tekinti elsőként, amely egy szomszédsági reláció meglétét feltételezi a keresési térben. Ez a megközelítés vezetett a helyi keresési eljárások kifejlődéséhez [1; 8; 10; 23; 29; 31]. Általában ezek az eljárások valamilyen heurisztikát alkalmaznak egy új lehetséges megoldás generálására egy már meglévőből, annak környezetében. Habár ezek a heurisztikák jelentősen javíthatják a meglévő lehetséges megoldást, általában nem

[1] A 0-ák pontokkal lettek helyettesítve az áttekinthetőség kedvéért.
[2] Nincs él a halmaz elemei között.



2. ábra. Élhozzáadási és csúcsösszehúzási lépések egy Zykov-fában. Ugyanazon szín v_2 és v_5-höz való rendelése nem javallott, mivel az nem optimális színezéshez vezet ezért köztük egy él behúzása egy ésszerű lépés. Továbbá ugyanazon szín rendelése a v_2 v_6 csúcsokhoz viszont támogatja a minimális színezést, ezért ezek összehúzhatók.

szolgáltatják az optimumot, így ezek a közelítő algoritmusok kategóriájába esnek. Manapság számos vizsgálat történt a sztochasztikus folyamatok alkalmazására a helyi kereső algoritmusokban, amelyek javíthatják azok hatékonyságát. Egy példa ezen eljárásokra a tabu-keresés [3; 31], szimulált hűtés [9; 33] és a hangya kolóniákkal való optimalizálás [5; 12]. A sztochasztikus eljárások közül sok a lehetséges megoldások egy populációját tartja számon és hasznosítja. Ilyenek az az evolúciós algoritmusok, amelyek alkalmazása népszerű a gráfszínezési algoritmusok között [2; 13; 14; 19; 20; 22; 25; 30; 48; 56; 59]. A színezési algoritmusok között számos egészértékű programozási megközelítéssel is találkozhatunk, ahol az egészértékű változók bizonyos struktúrákat azonosítanak például: független csúcshalmazokat [50]; színeket és csúcsokat [11; 51; 53]; aciklikus gráf orientációt [21]. Számos esetben bináris változókról van szó, amelyek bináris vektorként összefoghatók. Így az optimális megoldás(ok) is egy bináris vektorként áll(nak) elő megalkotva az ún. színezési politópot. Ezek a politópok állnak a probléma vizsgálatának a középpontjában [6; 24]. Számos relaxált verziója létezik az egészértékű feladatoknak, amelyek egy közelítését kívánják megadni a politóp egy-egy lapjának [18; 44; 50; 52; 57]. Sajnos ezen egészértékű feladatok megoldása jelentős időt vehet igénybe, ezért különböző technikákat fejlesztettek ki, hogy a futási sebességét javítsák, ilyen például az oszlopgenerálás módszere a korlátozás és szétválasztási módszerrel karöltve [7; 50; 53; 57]. Valójában itt a korlátozás és szétválasztás módszer implicit módon Zykov ötletén tartalmazza ([57]). Az elmúlt évszázad közepén Zykov összekötetlen csúcsok összehúzását és élhozzáadást alkalmazott gráfokon színhozzárendelés helyett, a színezési problémára. Egy példa erre a 2. ábrán látható. Ezekkel a műveletekkel új gráf jön létre mely örökölheti az eredeti gráf színezési tulajdonságait, mint például a kromatikus számát.

A disszertációban a szerző Zykov ötletéből kiindulva bevezetett négy különböző modellt (Merge Modellt). Ezen modellek hatékonyságát mutatta be elméleti és kísérleti eredményeken keresztül. A Merge Modellek az eredeti színezési probléma ekvivalens átfogalmazásai. Három ezeken a modelleken nyugvó általános színezési keretrendszert (Merge Keretrendszerek) definiált a szerző amelyek általánosításai a hagyományos szekvenciális színezési sémáknak. A modellek létező és új színezési algoritmusoknak egy egységes és tömör leírását teszik lehetővé. Amely számos előnnyel jár: egységes strukturális összehasonlítás, futási idő csökkenés, színezési teljesítménybeli javulás valamint új algoritmikus megközelítések támogatása. A szerző bemutatta a módszer hatékonyságát és előnyeit. A szerző megalkothatott számos új (színezési) Merge Stratégiát és Merge Algoritmust amelyek hatékonynak bizonyultak az elméleti elemzés és tapasztalati vizsgálatok során. Továbbá kidolgozott egy általános módszert amellyel általános stratégiák tervezhetők és mesterséges intelligenciai eljárások alkalmazhatók a színezési probléma megoldásában.

Kvóciens és Hatvány Módszer

A szerző a gráfszínezési folyamatot kvóciens és hatványgráfok segítségével, gráf homomorfizmusokon sorozatával definiálta (lásd Juhos et al. [37; 41]), amint azok a 3. ábrán láthatók. A homomorfizmusok az azonos színű csúcsok következetes összehúzásából vagy csoportba foglalásából származnak. A szerző Kvóciens és Hatvány Módszernek nevezte el az ezen elven alapuló színezési módszereit. Ezeknek célja egy olyan homomorfizmus megtalálása amely az eredeti gráfot[3] egy megfelelő teljes gráfba vagy azzal homomorf gráfba képezi. Az így kapott homomorfizmus meghatároz egy színezést az eredeti gráfra. A szekvenciális színezési eljárások támogatása végett a tekintett homomorfizmus további homomorfizmusok egymásutánjaként, kompozíciójaként kerül előállításra, megadva egy ún. közbenső homomorfizmus sorozatot. Ezen homomorfizmusok hasznos közbenső gráf struktúrákat hoznak létre, amelyek vizsgálata hatékony színezési eljárásokat eredményeztek valamint a színezési folyamatba egy alternatív betekintést nyújtanak (lásd Juhos et al. [37–43]).

Merge Modellek

Gráf homomorfizmusok definiálják a kapcsolatot az eredeti és egy kvóciens vagy hatvány gráf/multigráf között. A szerző definiált négy mátrix műveletet, amelyeket Merge Műveleteknek, vagy röviden Merge-nek nevezett el (lásd Juhos et al. [37; 41]). Egy Merge Művelet az eredeti gráf szomszédsági mátrixát képezi le egy mátrixba amely egy kvóciens gráf/multigráfot vagy hatvány gráf/multigráfot határoz meg, ezeket a szerző Bináris/Integer Merge Square-nek ($A/\mathbb{A}$) és Bináris/Integer Merge Table-nek ($T/\mathbb{T}$), vagy összefoglaló nevükön Merge Mátrixoknak nevezett el [37; 41]. Általánosságban egy Merge Mátrixot M fog a következőkben jelölni. Egymást követő Merge Műveletek sorozata hoz létre egy hagyományos értelemben vett színezést. A Merge

[3] Vagy egy ekvivalens átalakítását.

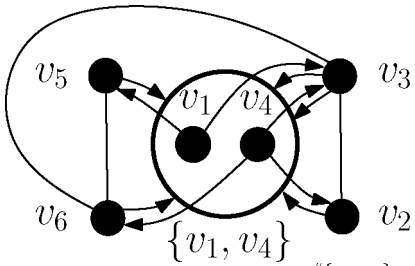
feltétele[4] $M_{ij} = 0$, ami azt jelenti, hogy M_i és M_j sorok (és oszlopok is, ha szükséges) Mergelhetők, amint az 1(b) ábra mutatja. Az utolsó lehetséges Merge-vel létrejövő záró homomorf kép (Merge Mátrix) sorai határozzák meg a színezés egy lehetséges megoldását. Az illető (Merge-elt) sorok független csúcshalmazokhoz tartoznak, ezek határozzák meg a színosztályokat. A 3. ábra példákat mutat a Merge Műveletekre, míg az 1. táblázat definiálja a műveleteket sor és mátrix műveletek segítségével is. Továbbá ezen táblázat összefüggéseket tartalmaz a Bináris Merge Mátrixok A és T és Integer párjaik $\mathbb{A}$ és $\mathbb{T}$ között.

Sor műveletekkel		Mátrix műveletekkel
$\mathbb{T}_i^{[t+1]} = \mathbf{a} + \mathbf{b}$	$\mathbb{T}_j^{[t+1]} = 0$	$\mathbb{T}^{[t+1]} = (I+W)\mathbb{T}^{[t]}$
$T_i^{[t+1]} = \mathbf{a} \vee \mathbf{b}$	$T_j^{[t+1]} = 0^T$	$T^{[t+1]} = T^{[t]} \vee PT^{[t]} - MT^{[t]}$
$T_i^{[t+1]} = \mathbb{T}_i^{[t+1]} - \mathbf{a} \circ \mathbf{b}$	$T_j^{[t+1]} = 0^T$	$T^{[t+1]} = \mathbb{T}^{[t+1]} - \sum_j (\mathbf{a} \otimes \mathbf{b})(I_j \otimes I_i)$
$\mathbb{A}_i^{[t+1]} = \mathbf{a} + \mathbf{b}$	$\mathbb{A}_j^{[t+1]} = 0^T$	$\mathbb{A}^{[t+1]} = (I+W)\mathbb{A}^{[t]}(I+W)^T$
$\mathbb{A}_{\_i}^{[t+1]} = \mathbf{a}^T + \mathbf{b}^T$	$\mathbb{A}_{\_j}^{[t+1]} = 0$	
$A_i^{[t+1]} = \mathbf{a} \vee \mathbf{b}$	$A_j^{[t+1]} = 0^T$	$A^{[t+1]} = A^{[t]} \vee (PA^{[t]}P^T) - (MA^{[t]}M^T)$
$A_i^{[t+1]} = \mathbb{A}_i^{[t+1]} - \mathbf{a} \circ \mathbf{b}$	$A_j^{[t+1]} = 0^T$	
$A_{\_i}^{[t+1]} = (A_i^{[t+1]})^T$	$A_{\_j}^{[t+1]} = 0$	

1. táblázat. A különböző Merge Műveletek meghatározásai több módon. A $\mathbf{a}$ és $\mathbf{b}$ az M Merge Mátrix i-edik és j-edik Merge-lhető sorát jelölik. A $^{[t]}$ felső index a t-edik Merge lépést jelenti $\mathbf{a}$ és $\mathbf{b}$ sorok Merge-e esetén; a $P = I_i \otimes I_j$, $R = I_j \otimes I_j$, $W = P - R$, ahol I_i az i-edik sora az egységmátrixnak I-nek. M_i az M mátrix i-edik sora, míg $M_{\_i}$ az i-edik oszlopot jelöli. A $\circ$ a Hadamard-Schur szorzatot, valamint $\otimes$ a diadikus szorzatot jelöli.

A Merge Table-k az eredeti gráf csúcsai és a szomszédos színosztályok közötti kapcsolatot írják le, azaz hogy van-e él közöttük vagy sincs. Alapvetően az eredeti gráf szomszédsági mátrixa hozzárendelhető egy színosztályhoz (amely egy hatvány gráf csúcsa), valamint az oszlopok az eredeti gráf csúcsaihoz. A korábbiak szerint két altípust különböztetünk meg, nevezetesen a súlyozottat (Integer Merge Table Model/hatvány multigráf) és a súlyozatlant (Bináris Merge Table Model/hatvány gráf), annak megfelelően, hogy a Merge folyamat során keletkező többszörös éleket figyelembe vesszük vagy sem. Ezek két alapvető sor műveletetet határoznak meg az összeadás és az elemenkénti bináris OR műveletet. Ha csak a sorokra alkalmazzuk őket, akkor hatvány multigráfokat/gráfokat, azaz Integer/Bináris Merge Table-ket kapunk, ha az oszlopokra is alkalmazzuk egyben akkor kvóciens multigráfokat/gráfokat, azaz Integer/Bináris Merge Square-eket kapunk. A Merge Square-knél mind a sorok, mind az oszlopok színosztályokat azonosítanak és azok közötti viszonyokat írnak le, azaz hogy van-e él közöttük vagy sincs. Minden Merge Mátrix közös jellemzője, hogy egy sor az eredeti gráf egy független csúcshalmazát határozza meg. Itt jegyezzük meg, hogy a kezdeti Merge Mátrix a szomszédsági mátrix, melynek sorai a csúcsokat azonosítják. Emlékezzünk, a színosztályok független csúcshalmazok, valamint a csúcsok egy elemű független csúcshalmazokat határoznak meg. A Merge Műveletek hagyományos értelemben ezek unióját

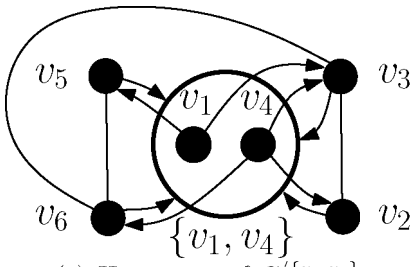
[4] M_{ij} az (i,j)-edik mátrixelem.



(a) Hatvány multigráf $G^{/\!/\{v_1,v_4\}}$

	v_1	v_2	v_3	v_4	v_5	v_6
$\{r_1, r_4\}$	0	1	2	0	1	1
r_2	1	·	1	·	·	·
r_3	1	1	·	1	·	1
r_5	·	·	·	1	·	1
r_6	1	·	1	·	1	·

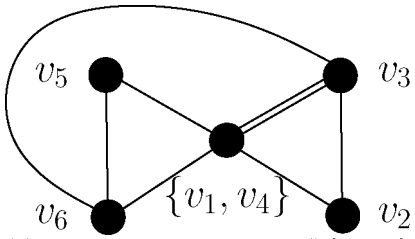
(b) Integer Merge Table. Az r_1 és r_2 sorok összeadódtak.



(c) Hatvány gráf $G^{/\{v_1,v_4\}}$

	v_1	v_2	v_3	v_4	v_5	v_6
$\{r_1, r_4\}$	0	1	1	0	1	1
r_2	1	·	1	·	·	·
r_3	1	1	·	1	·	1
r_5	·	·	·	1	·	1
r_6	1	·	1	·	1	·

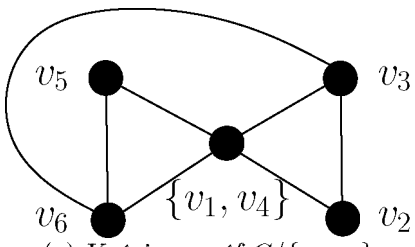
(d) Binary Merge Table. Az r_1 és r_2 sorokon elemenkénti OR művelet lett végrehajtva.



(e) Kvóciens multigráf $G /\!/ \{v_1, v_4\}$

	$\{v_1, v_4\}$	v_2	v_3	v_5	v_6
$\{r_1, r_4\}$	0	1	2	1	1
r_2	1	·	1	·	·
r_3	2	1	·	·	1
r_5	1	·	·	·	1
r_6	1	·	1	1	·

(f) Integer Merge Square. Az r_1 és r_2 sorok és v_1 és v_4 oszlopok összeadódtak.



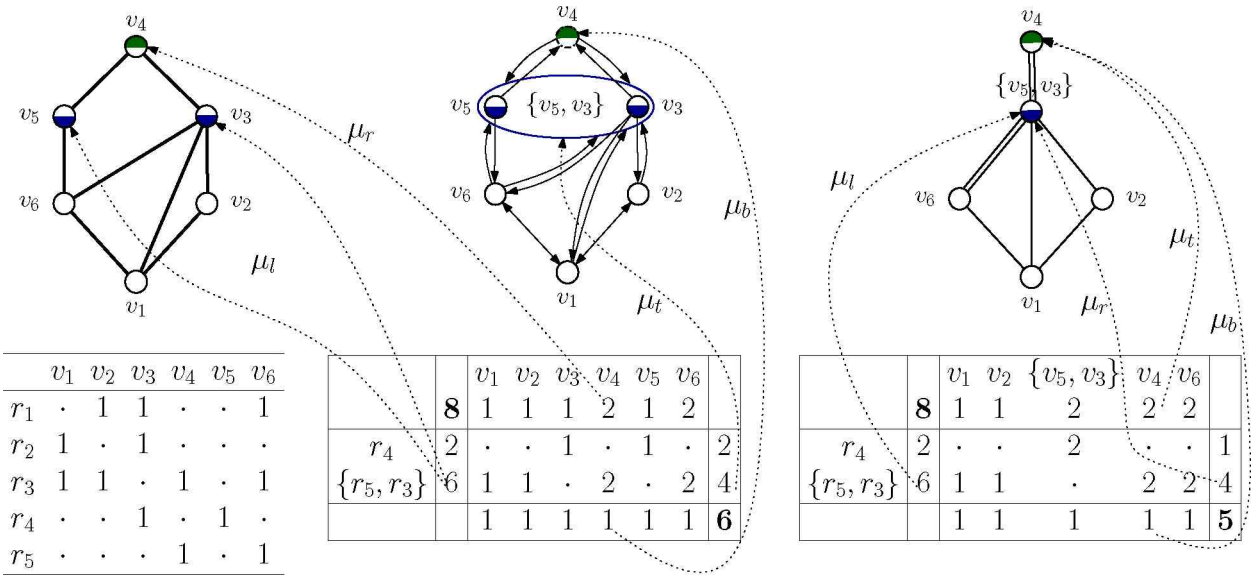
(g) Kvóciens gráf $G/\{v_1, v_4\}$

	$\{v_1, v_4\}$	v_2	v_3	v_5	v_6
$\{r_1, r_4\}$	0	1	1	1	1
r_2	1	·	1	·	·
r_3	1	1	·	·	1
r_5	1	·	·	·	1
r_6	1	·	1	1	·

(h) Binary Merge Square. Az r_1 és r_2 sorokon és v_1 és v_4 oszlopokon elemenkénti OR művelet lett végrehajtva.

3. ábra. Különböző Merge Műveletek eredményei.

jelentik. Ezen modelljét a színezésnek a szerző Merge Modellnek nevezte el. A 4. ábra a Merge Mátrixok különböző struktúráit/részstruktúráit mutatja utalva a megfelelő gráfra/részgráfra. A



	v_1	v_2	v_3	v_4	v_5	v_6
r_1	·	1	1	·	·	1
r_2	1	·	1	·	·	·
r_3	1	1	·	1	·	1
r_4	·	·	1	·	1	·
r_5	·	·	·	1	·	1

		v_1	v_2	v_3	v_4	v_5	v_6	
	8	1	1	1	2	1	2	
r_4	2	·	·	1	·	1	·	2
$\{r_5, r_3\}$	6	1	1	·	2	·	2	4
		1	1	1	1	1	1	**6**

		v_1	v_2	$\{v_5, v_3\}$	v_4	v_6	
	8	1	1	2	2	2	
r_4	2	·	·	2	·	·	1
$\{r_5, r_3\}$	6	1	1	·	2	2	4
		1	1	1	1	1	**5**

4. ábra. Az eredeti gráf és egy rész színezése a kezdeti Merge Mátrixszal, a szomszédsági mátrixszal; egy a színezésnek megfelelő hatvány multigráf és a színezett csúcsokhoz tartózó Integer Merge Table részmátrixszal; egy a színezésnek megfelelő kvóciens multigráf és a színezett csúcsokhoz tartózó Integer Merge Square részmátrix. A segédstruktúrák a részmátrixok oldalain helyezkednek el. Ezek összegzik vagy számolják a nem-zéró elemeket a sorokban és az oszlopokban. A μ_l bal sorösszeg az eredeti gráfban a megfelelő csúcsok fokszám összegét adja; a μ_r jobb sorösszeg egy színosztály szomszédjainak a számát adja; a μ_t a szomszédos színezett csúcsok számát adja; valamint μ_b a szomszédos színek számát mutatja.

modell támogatja a párhuzamos szoftver és hardver implementációt. Egy szekvenciális színezési algoritmus amely ezen modellre épül jelentős teljesítménybeli javulást könyvelhet el. A szerző ezen javulást elméletileg és tapasztalatilag is alátámasztotta (lásd Juhos et al. [40]) valamint hatékony új színezési eljárásokat dolgozott ki ezen modellek segítségével [35; 38–40; 42; 43].

Merge Keretrendszerek

A Merge Modellek a gráfszínezést mátrix reprezentáció és speciális műveletek útján definiálják. A szerző kidolgozott három általános keretrendszert amelyek absztrakt színezési algoritmusokat határoznak meg (lásd Juhos et al. [41; 42]). Ezen absztrakciók az általánosításai a tradicionális színezési sémáknak. A Merge Műveletek helyettesítik a hagyományos értelemben vett színezést. A Merge Modellekben eltűnik a különbség a szín és a csúcs kiválasztási stratégiák között. Elegendő egy általános sorválasztási stratégiát meghatározni, amely alkalmas színezett vagy színezetlent sorok kiválasztására is, ha a tradicionális színezési sémákat akarjuk követni. Azonban itt a színek csak jelzés értékűek, azt jelzik, hogy egy sor érintett volt-e már a Merge Műveletben. Attól függően, hogy milyen sorrendben választjuk ki a különböző állapotú (színezett/színezetlen)

sorokat kaphatunk két eltérő keretrendszert: vagy először egy színezetlen (Uncoloured) sort választunk, majd egy színezettet (Coloured) a Merge Művelethez (UC Merge Keretrendszer) vagy fordítva (CU Merge Keretrendszer). Ezen keretrendszerek általánosításai a hagyományos színezési sémának (lásd Juhos et al. [41]). Ezen keretrendszerek pszeudokódját a 5. ábra ismerteti. A choose-unc és choose-col eljárás/kiválasztási stratégiák nem pontosan definiáltak, absztrakt

UC MERGE KERETRENDSZER(A szomszédsági mátrix)
1 $M \leftarrow A$
2 **repeat**
3 $u \leftarrow \arg \text{choose-unc}_i\{M_i^{unc}\}$ //Nem színezett sor (index) választás.
4 $c \leftarrow \arg \text{choose-col}_i\{M_i^{col}\}$ //Színezett sor (index) választás [a] ahol $M_{uc} = 0$
5 $M \leftarrow \text{merge}(M, \{u, c\})$ //Az u és c sorok/oszlopok Merge-elése. [b]
6 **until** M^{unc} üres
7 **return** M

CU MERGE KERETRENDSZER(A szomszédsági mátrix)
1 $M \leftarrow A$
2 **repeat**
3 $c \leftarrow \arg \text{choose-col}_i\{M_i^{col}\}$ //Színezett sor (index) választás.
4 $u \leftarrow \arg \text{choose-unc}_i\{M_i^{unc}\}$ //Nem Színezett sor (index) választás [c], ahol $M_{cu} = 0$
5 $M \leftarrow \text{merge}(M, \{u, c\})$ //Az u és c sorok/oszlopok Merge-elése.
6 **until** M^{unc} üres
7 **return** M

[a] $M_{uc} = M_{cu} = 0$ a Merge feltétel, azaz, hogy nincs él az adott független csúcshalmazok között.
[b] A Merge Square-eknél, az oszlopok is részt vesznek a Merge Műveletben.
[c] $M_{cu} = M_{uc} = 0$ a Merge feltétel, azaz, hogy nincs él az adott független csúcshalmazok között.

5. ábra. Az UC és CU Merge Keretrendszer

CC MERGE KERETRENDSZER(A szomszédsági mátrix)
1 $M \leftarrow A$
2 **repeat**
3 $\{i, j\} \leftarrow \arg \text{choose}_{\{i,j\}}\{M_i, M_j : i \neq j\}$ //Két sor (index) választása [a], ahol $M_{ij} = 0$
4 $M \leftarrow \text{merge}(M, \{i, j\})$ //Az i és j sorok/oszlopok Merge-elése.
5 **until** M nem Merge-lhető
6 **return** M

[a] $M_{ij} = M_{ji} = 0$ a merge feltétel, azaz, hogy nincs él az adott független csúcshalmazok között.

6. ábra. A CC Merge Keretrendszer

stratégiák. Ezek helyettesíthetők konkrét választási stratégiákkal amelyek a színezett M^{col} és nem színezett M^{unc} rész-Merge-Mátrixokat használják. Ezen mátrixok tartalmazzák a színezett M_i^{col} és nem színezett M_j^{unc} sorait az eredeti Merge Mátrixnak. A 4. ábra példákat mutat a szí-

nezett rész-Merge-Mátrixokra. A choose-unc eljárás kiválaszt egy színezetten sort/csúcsot, míg a choose-col egy színezett sort/'színosztályt' választ vagy egy üres sort képez a színezett rész-Merge-Mátrixokban, ami az egy operandusú Merge-t támogatja, ezzel nyitva egy egy elemű új színosztályt. Valójában nem szükséges megkülönböztetni a színezett és színezetlen státuszokat, egy közös kiválasztási stratégia a choose választhatna tetszőleges két sort egy Merge Mátrixból, hogy végrehajtsa rajtuk a Merge Műveletet. Ez a megközelítés a CC Merge Keretrendszerben lett definiálva (lásd Juhos et al. [37]), melynek pszeudokódját az 6. ábra mutatja. Egy sor a Merge Mátrixban egy színosztályt azonosít, azaz független csúcshalmazt. Hagyományos értelemben a CC Merge Keretrendszerben két tetszőleges színosztályt/független csúcshalmazt választunk majd ezek unióját képezzük. Míg az UC és CU Merge Keretrendszerekben a választott független csúcshalmazok közül az egyik mindíg egyelemű, ezel a színezettlen csúcsok, amelyeket a színezettlen sorok azonosítanak. A CC Merge Keretrendszer a legáltalánosabb, habár ez lefedi az UC és CU keretrendszereket, azok külön definíciója szükséges az algoritmusok egyszerűbb rendszerezése miatt, annál is inkább, mert a tradicionális színezési sémák ezek valamelyikébe illenek. Ezen keretrendszerek az új színezési modellel támogatják az egységes algoritmus analízist (lásd Juhos et al. [38; 40; 42; 43]).

A szerző bemutatta, hogy egy színezési algoritmus futási teljesítménye javul egy megfelelő Merge eretrendszerbe ágyazzuk egy megfelelő Merge Modellt alkalmazva. Anélkül, hogy az algoritmus lépéseit megváltoztatnák a Merge Modellbeli reprezentáció számítási teljesítménybeli igény csökkentést okoz. Eiben és van Hemert et al. rámutattak hogy az élek vizsgálata kulcsfontosságú faktor a legtöbb színezési algoritmus futása tekintetében [19; 59]. A tradicionális sémák esetén amikor egy csúcsot szeretnénk színezni, akkor két választásunk van az élvizsgálatokra amelyel a helyes színezést biztosítjuk: a színezett csúcsokat vesszük számba ($\mathcal{A}_{col}$), vagy az színezendő csúcs szomszédait vesszük számba ($\mathcal{A}_{neigh}$). A következőkben láthatjuk, hogy az élvizsgálatok jelentős mértékben csökkenthetők, ha alkalmazzuk a Merge Modelleket ($\mathcal{A}_{mm}$).

Corollary 1 ([40]) *Ha adott egy véletlen gráf $G_{n,p}$ fix p élvalószínűséggel és adott egy színezési algoritmus $\mathcal{A}$, akkor a következőkben alakul az élvizsgálatok száma $\#(.)$:*

1. *Színezett csúcsok vizsgálata esetén:* $\#(\mathcal{A}_{col}) = \mathcal{O}(n^2)$
2. *Szomszédok vizsgálata esetén:* $\#(\mathcal{A}_{neigh}) = \mathcal{O}(n^2)$
3. *Merge-lt sorok/színosztályok esetén:* $\#(\mathcal{A}_{mm}) \leq \mathcal{O}\left(\frac{n^2}{\log n}\right)$

A fenti tétel az aszimptotikus viselkedést mutatja, azonban megvizsgálhatjuk a legrosszabb esetet is.

Corollary 2 ([40]) *Legyen G egy tetszőleges gráf ekkor az alábbiak érvényesek*

1. $\#(\mathcal{A}_{mm}) \leq \#(\mathcal{A}_{col})$
2. $\#(\mathcal{A}_{mm}) \leq \#(\mathcal{A}_{neigh})$

Mindhárom keretrendszer egy egységes szerkezetet tükröz. Az algoritmusok ezen keretrendszerekben Merge Mátrix sorok sorozatos kiválasztását végzik, majd végrehajtanak rajtuk egy Merge Műveletet, mely eredményeképpen végül előáll egy színezés. Egyik általános keretrendszernek sincs konkrét sorkiválasztási stratégiája. A keretrendszerek konkrét kiválasztási stratégiákkal alkotnak algoritmusokat, Merge Algoritmusokat.

Merge Stratégiák

Ahhoz, hogy egy konkrét algoritmust kapjunk, az algoritmus lépéseit definiálni kell, azaz itt Merge Műveletek egy sorozatát kell megadni. Egy Merge Művelet két sort/oszlopot vesz az aktuális Merge Mátrixból és egy újabb Merge Mátrixot képez, ha a Merge feltétel engedi. Merge-k ismételt alkalmazása elvezet egy záró Merge Mátrixhoz, ahol további Merge nem lehetséges. A Merge-ek sorozata alapvető befolyással bír a lehetséges megoldás jósága tekintetében, ez határozza meg a használt színek számát, amely egybeesik a záró mátrix sorainak számával[5]. A szerző több Merge Stratégiát dolgozott ki, hogy hatékony Merge sorozatot állítson elő [35; 37–43]. Ezen stratégiák hasznosnak bizonyultak az elméleti és kísérleti elemzés folyamán is. Az új probléma megközelítés új stratégiák kialakítására ad lehetőséget, amint alább láthatjuk. Az egyszerűség kedvéért a felsorolt stratégiák Bináris Merge Modelleket feltételeznek, habár Integer megfelelőjük is tárgyalásra került a disszertációban, ahol az Integer Merge Modellek erényei külön kerültek tárgyalásra [40], néhányat említve belőlük: támogatják a visszalépést (backtracking) és a másodlagos döntéshozatalt (tie breaking), ha az elsődleges stratégia több sort is kiválasztana Merge-elésre. A következő stratégiák sor-pár kiválasztási stratégiák, azaz a legáltalánosabb keretrendszerhez, a CC Merge Keretrendszerhez alkalmas választási eljárások, de az UC es CU Keretrendszerekhez is használhatók. Legyen a sor-pár kiválasztás stratégia alapja az $\hat{X}$ mátrix. A mátrix $\hat{X}_{ij}$ eleme legyen arányos az M_i és M_j sorok kiválasztási valószínűségével egy algoritmus minden lépésében[6]. A sor-pár kiválasztási stratégia válassza a következő két sort $\{i,j\} = \arg\max_{ij} \hat{X}_{ij}$. A következő stratégiák meghatározzák $\hat{X}$-et egy tetszőleges Merge Mátrixra, így egy algoritmus minden lépését definiálják.

A leghosszabb Merge sorozat. Mivel a Merge Mátrix sorok M_i-k színosztályokat azonosítanak. Ennélfogva a cél a sorok számának csökkentése. Ezt a leghosszabb Merge sorozattal létrehozásával érhetjük el, mivel a sorok száma minden lépésben eggyel csökken. Ennek érdekében a szerző bevezetett két stratégiát (lásd Juhos et al. [38]). A Dot Product Stratégia a nem-zéró elemek alakulását követi nyomon a Merge-k során. Megkísérli azok számát minimálisan tartani, mivel azok akadályozzák a Merge-eket, így a stratégia alapja

$$\hat{X}_{ij} = \langle M_i, M_j \rangle \, [M_{ij} = 0] \tag{1}$$

ahol $[M_{ij} = 0]$ a Kronecker delta függvény, mely $[x = x] := 1$, egyébként 0 (ez magába foglalja

[5]Emellett, a Merge Table-kban a zeró elemek a színosztályok elemeit azonosítják egy optimális megoldás esetén, ha színezés szempontjából jelentéktelen alacsony fokszámú csúcsok $\{v : d(v) < \chi\}$ el lettek távolítva.

[6]$\hat{X}$ minden lépésben változhat.

a Merge feltételt) M_i és M_j a i-edik és j-edik sor egy Bináris Merge Mátrixban[7]. Egy A Bináris Merge Square-re, ez az alábbiak szerint írható

$$\hat{X} = AA^T \circ \bar{A} \tag{2}$$

Bár a nem-zéró elemek meggátolhatják a Merge Műveleteket, a zéró elemek segítenek, támogatják azokat. Így a Cosine Stratégia figyelembe veszi mindkettő alakulását a Merge-k során és annak megfelelően alakítja a sor kiválasztásokat, az alábbiak szerint

$$\arg\max_{i,j} \hat{X}_{ij} = \arg\max_{i,j} \frac{\langle M_i, M_j \rangle}{|M_i|\,|M_j|}[M_{ij} = 0] = \arg\max_{i,j} \frac{\langle M_i, M_j \rangle}{||M_i||\,||M_j||}[M_{ij} = 0] \tag{3}$$

Párhuzamos sorok. A Cosine Stratégia előnyben részesíti a párhuzamos sorokat a Merge Mátrixokban. Ez ésszerű választás azért is mert a szomszédsági mátrix sorai amelyek azonos színosztályhoz tartoznak egy optimális színezésben majdnem párhuzamosak. A Merge-k során a keletkező Merge Square Mátrixokban ez a párhuzamos tulajdonság egyre karakteresebbé válik. A Merge Square-ek ésszerű módosításai Karger et al. [44] munkássága nyomán további támogatást nyújt a Cosine stratégia számára. Felhasználva ezt az szerző definiálta a 'Zykov-fa és Lovász-theta' stratégiát (lásd Juhos et. al. [35; 43])

Szín hasonlóság. Valójában a Zykov-fa és Lovász-theta stratégia a csúcsok szín hasonlóságának becslésén alapszik. A szomszédsági mátrix egy szín különbözőségi relációt határoz meg, mivel a csúcsok amelyek (él-)relációban vannak nem színezhetők azonosan. Ennek ellenkezője a színezési reláció. Egy színezés megadható egy szín hasonlósági reláció meghatározásával, itt csak az azonosan színezett csúcsok állnak relációban. A reláció egy $\{0,1\}-$mátrixszal kifejezhető, ez a színezési mátrix. Ez megadja, hogy két csúcs azonos színű vagy különböző. Bár az optimális színezések mátrixa is megadható eképpen, ezekre nem támaszkodhatunk, mert ezek alkotják a feladat megoldását. Noha ezek nem ismertek, az átlaguk közelíthető egy szemi-definit program (lásd Karger et al. [44]) megoldásával amely a Lovász-theta-t is szolgáltatja eredményül $\bar{\theta}$:

$$\bar{\theta} = \min_t \{t : Z \succeq 0, z_{ii} = t - 1, z_e = -1 \; \forall e \in E\} \tag{4}$$

Így egy közelített szín hasonlósági relációt kapunk. Ez egy Z_{opt} valós értékű mátrixszal írható le, amely egy megoldása az Eq. 4-nek. Legyen $\hat{X}$ az alábbiak szerint származtatva

$$\hat{X} = (Z_{opt} + 1) \circ (1 - I)$$

ahol I az egységmátrix. A legnagyobb és legkisebb elemei $\hat{X}$-nek fontos információt hordoznak. A szerző ezen információkat valamint Zykov munkásságát felhasználva (lásd [64; 65]) elkészítette a 'Zykov-fa és Lovász-theta' stratégiát [35; 43]. Ahol egy kvóciens gráf csúcsai összekötendők ($\arg\min_{i,j}\{\hat{X}_{ij} : \hat{X}_{ij} < 0\}$ csúcsok) vagy Merge-lendők ($\arg\max_{i,j} \hat{X}_{ij}$ csúcsok) a kicsi illetve nagy közelített hasonlósági értékeknek megfelelően (lásd 2. ábra). Az összekötési (él hozzá-

[7] Bináris Merge Mátrixok esetén $[M_{ij} = 0] \equiv (1 - M_{ij})$.

adási) és Merge Műveletek során a hasonlóság egyre karakterisztikusabbá válik, támogatva ezzel az egyre értékesebb sor kiválasztásokat. Az algoritmus több él ($\{ij : \hat{X}_{ij} < 0\}$ élek) együttes hozzáadásával vagy több Merge művelet (a $\{\{i,j\} : \hat{X}_{ij} > 0.5\bar{\theta}\}$ csúcsokon) elvégzésével felgyorsítható, mivel noha a szemi-definit probléma Eq. 4 polinomiális időben megoldható, a gyakorlatban azonban ez időigényes.

Norma minimalizálás az eredményben. A Dot Product Stratégia azt a két sort M_r és M_s választja ki Merge-elésre, amelyeknek maximális a skaláris szorzatuk $\{r,s\} = \arg\max_{i,j} \langle M_i, M_j \rangle$. Ez az eredmény Merge Mátrixban az elemenkénti $1-$norma[8] minimalizálását eredményezi, mivel $\left|M_{/rs}\right| = |M| - \max_{i,j} \langle M_i, M_j \rangle = |M| - \langle M_r, M_s \rangle$, így

$$\arg\left(|M| - \max_{i,j} \langle M_i, M_j \rangle\right) = \arg\min_{i,j} \left(|M| - \langle M_i, M_j \rangle\right) = \arg\min_{i,j} \left|M_{/ij}\right| \tag{5}$$

A záró Merge Mátrix, ami egy optimális megoldáshoz tartozik, rendelkezik a legkisebb elemenkénti mátrixnormával az összes lehetséges záró mátrix közül. Emiatt az elemenkénti norma minimalizálása ésszerű stratégia. Továbbá egy optimális zárómátrixnál a származtatott mátrixnormák is minimálisak. Ez a megfigyelés vezetett a szerző legnagyobb norma csökkentés stratégiájához (lásd Juhos et al. [42]). Speciális esetben ez a Spektrálnorma minimalizálási stratégiához vezet, amely a legkisebb a származtatott normák között és ennélfogva jó karakterizációja egy mátrixnak. A szerző a spektrálnorma minimalizálási stratégiát elemezte, amely hatékonynak bizonyult az elemzések során. Mivel a spektrálnorma a származtatott 2-normával egyezik, ezért ezen stratégia alapja a következő

$$\hat{X}_{ij} \quad = \quad \begin{cases} \frac{1}{||M_{/ij}||_2} [M_{ij} = 0] & i \neq j \\ 0 & i = j \end{cases} \tag{6}$$

A Spektrálnorma Stratégia próba Merge-ket kell, hogy végezzen, ahol az eredmény mátrix normája határozza meg a kiválasztási stratégiát. Ez számításigényes feladat. Merikoski és Kumar megadott több hatékony spektrálnorma közelítési formulát. (lásd [54]). Legyen $M = A$ egy Binary Merge Square ekkor egy közelítés

$$||A_{/ij}||_2 \approx \sqrt{\frac{\sum_{r=1}^{l} \langle (A_{/ij})_r, \mathbf{e} \rangle^2}{l}} \tag{7}$$

ahol l a sorok száma az $A_{/ij}$ 'próba' Merge Mátrixban. Felhasználva ezen formulá(ka)t a szerző adaptálta a Spektrálnorma stratégiát és közelített Spektrálnorma Stratégiákat vezetett be (lásd Juhos et al. [42]). A közelítéssel lehetőség nyílik a választási stratégia közvetlen meghatározására

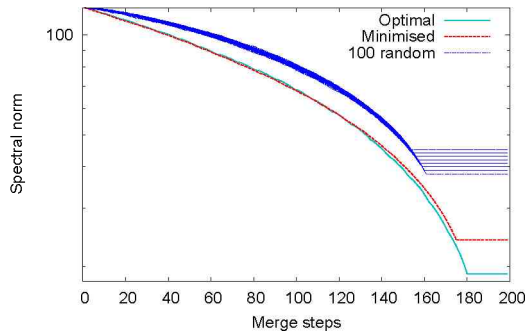
[8] Valójában az összes elemenkénti norma minimalizálását.

az aktuális Merge Mátrixból próba Merge-ek nélkül:

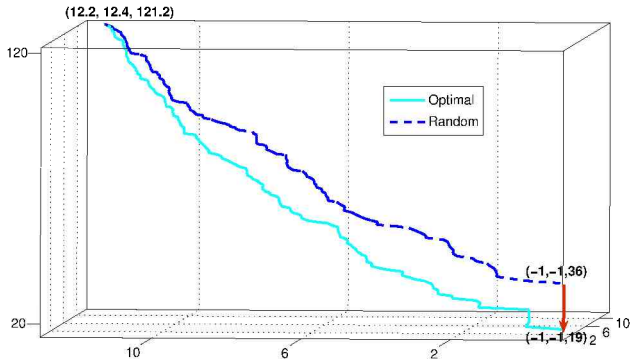
$$\begin{aligned}\langle (A_{/ij})_i, \mathbf{e}\rangle &= \langle A_i, \mathbf{e}\rangle + \langle A_j, \mathbf{e}\rangle - \langle A_i, A_j\rangle \\ \langle (A_{/ij})_j, \mathbf{e}\rangle &= 0 \\ \langle (A_{/ij})_r, \mathbf{e}\rangle &= \langle A_r, \mathbf{e}\rangle - 1 \quad r \in \mathcal{I} \\ \langle (A_{/ij})_r, \mathbf{e}\rangle &= \langle A_r, \mathbf{e}\rangle \quad r \notin \mathcal{I} \cup \{i, j\}\end{aligned} \tag{8}$$

ahol $\mathcal{I}$ egy index halmaz a A_i és A_j sorok közös egyeseinek pozíciójával. Így az Eq. 7 direktben számítható az aktuális Merge Mátrixból 'próba' Merge-k nélkül. Továbbá a közelítő formula rámutat a Dot Product Stratégiával való hasonlóságra. Az Eq. 7 egy hatékony stratégiát alkot egy T Bináris Merge Table-val is, de itt a direkt számításban a harmadik sora az Eq. 8-nak másként alakul: $\langle (T_{/ij})_r, \mathbf{e}\rangle = \langle T_r, \mathbf{e}\rangle$. Habár, ahhoz hogy az eredeti (közelítő) Spektrálnorma Stratégiát visszakapjunk Binary Merge Table-ra az Eq. 7[9], a $T_{/ij}T_{/ij}^T$ szimmetrikus mátrixon kell, hogy végbemenjen. Ekkor a $T_{/ij}$ 'próba' mátrix spektrálnorma négyzetét vagy annak közelítését kapjuk vissza, amely szintén alkalmas a stratégia megvalósítására, valamint ez esetben is fennáll a direkt kalkuláció lehetősége, a próba Merge-k és mátrixok elhagyása mellett.

Mátrix tulajdonságok – Merge Útvonal. A szerző bevezette a Merge Path-ok fogalmát [42]. A Merge-k nyomán a mátrixok tulajdonságai követhetők. A kívánt mátrix tulajdonságokból (pl. sajátértékek) alkossunk egy tulajdonságvektort. Képezzék ezek a vektorok az alapját a kiválasztási stratégiáknak (mint például a legnagyobb saját vagy szinguláris érték, azaz a spektrálnorma). Az egymást követő tulajdonságvektorok egy útvonalat, a Merge útvonalat határoznak meg. Az útvonal elemei összefüggésben vannak a színezés lépéseivel az útvonal vége pedig a színezés jóságával, amint azt a 7(a)). ábra is mutatja. Az ideális útvonal amely optimális színezéshez



(a) Legnagyobb spektrálnorma csökkentés. A görbék vége meg lett hosszabbítva a könnyebb összehasonlíthatóság miatt (vízszintes vonalak).



(b) Egy 3D Merge Útvonal, amely a három legnagyobb sajátérték útját mutatja a Merge-ek során.

7. ábra. A sajátértékek alakulása a Merge-k során. A tekintett gráf 20−kromatikus, egyenlően particionált, 200 csúcsú, 0.64 élsűrűséggel amely az ún. 'phase transition' terület közepén helyezkedik el. A spektrálnorma értéke a záró Bináris Merge Square esetén $\chi - 1 = 19$, ha az optimumot elértük, egyébként ez nagyobb.

[9] Vagy az eredeti spektrálnorma számítás.

vezethet nem ismert, mert a feladat egy ilyen útvonal megtalálása. Az optimális útvonal kezdő és a végpontjai általában ismertek, a szerző bevezetett egy általános stratégiát amely az optimális Merge Útvonal közelítésén alapszik (lásd Juhos et al. [42]) felhasználva egy előzetes tudást. Az előzetes tudás megszerzéséhez a szerző vázolt egy koncepciót a Merge Útvonal intelligens tanulási és klaszterezési eljárásokkal való ötvözésére (Juhos et al. [36] alapján).

Kiterjesztett heurisztikák és meta-heurisztikák. A szerző a nem Merge alapú színezési stratégiák egy Merge kiterjesztését határozta meg, az illető stratégiák egy megfelelő Merge Keretrendszerbe való beágyazásával, amelyek jelentős színezési teljesítmény javulást okoztak (lásd Juhos et al. [38]). A kiterjesztet stratégiák teljesítményének elméleti és tapasztalati vizsgálata javulást mutatott az eredetihez képest. A kiterjesztés a Bináris Merge Square-ek példáján egyszerűen nyomon követhető, habár általában a kiterjesztés a többi Merge Modellben is érvényes. Egy Bináris Merge Square a szomszédsági mátrixa egy kvóciens gráfnak, amely egy egyszerű gráf az eredeti gráfhoz hasonlóan. Így egy stratégia amely az eredeti gráf szomszédsági mátrixán működik, az egy Merge Square Modellel is képes együttműködni. Ez lehetőséget biztosít egy dinamikus felülvizsgálati eljárásra amely során minden Merge Művelet után, a stratégia képes előző döntéseit megváltoztatni, azon új információk alapján amely a keletkező Merge Mátrixban elérhető.

A Merge Modellek strukturális jellemzői támogatást nyújtanak meta-heurisztikák tervezéséhez is (lásd Juhos et al. [37]). A szerző gráfszínezési evolúciós algoritmusok számára definiált a Merge Table modellek segítségével egy finomított fitnesz függvényt a hagyományosan alkalmazott fitnesz javításaként. Melynek eredményeként egy simább optimalizálási felületet kapunk, növelve az optimalizálás hatékonyságát. A $\zeta_{M(\pi)}$ jelölje a nem-záró elemek számát a záró Merge Tableben (lásd 4. ábra), az új ζ-fitness függvény ekkor $f(\pi) = (k_{M(\pi)} - \chi)\zeta_{M(\pi)}$, ahol $M(\pi)$ az a záró Merge Table amely a π permutációhoz és egy mohó színezéshez tartozik. Ez követi az elemenkénti norma optimalizálási stratégiát a Bináris Merge Table-ben (lásd Dot Product Stratégia). Továbbá a szerző egy mutáció operátort is definiált amely a színezésben a Merge Modellek alapján a problémás/nehezen színezhető csúcsokat előreveszi a színezési folyamatban.

Merge Algoritmusok

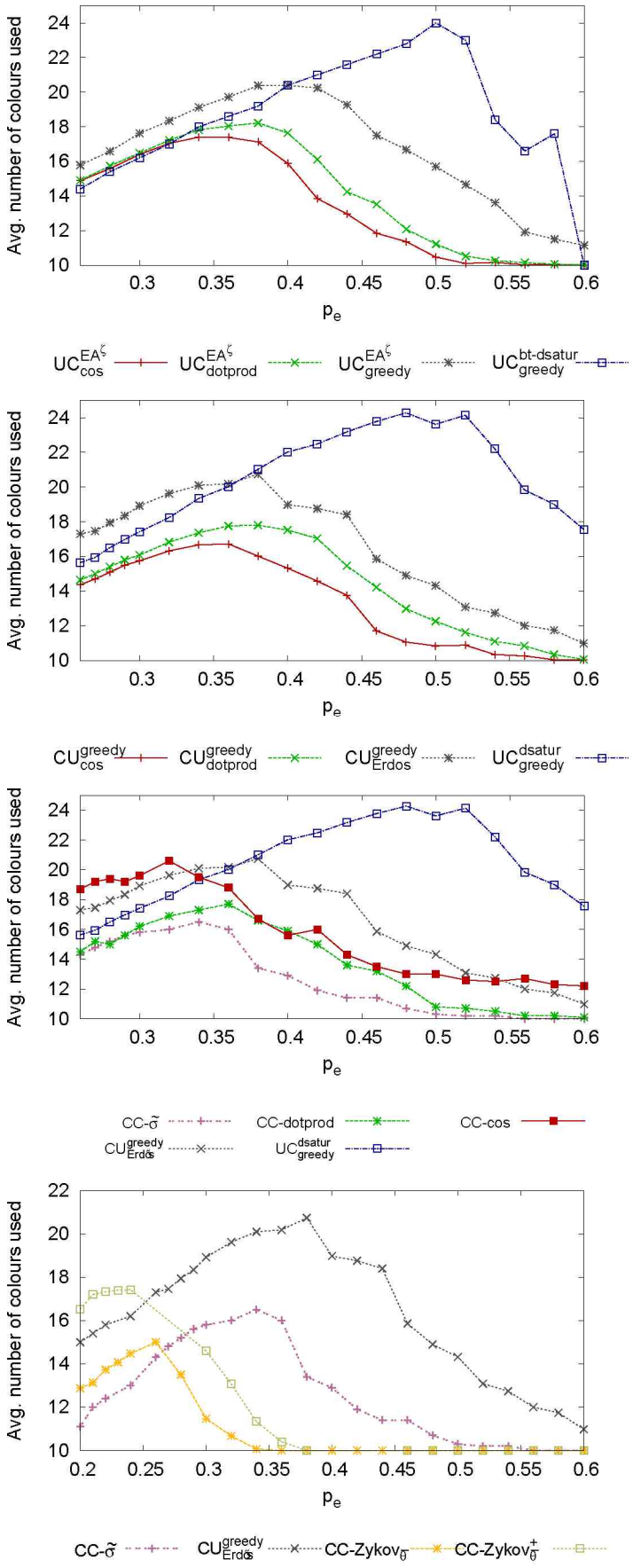
A szerző kombinálta a Merge Stratégiáit a különböző Merge Keretrendszereivel valamint elemezte ezek teljesítményét (lásd Juhos et al. [35; 37–43]). Az így keletkezett algoritmusok összehasonlításra kerültek standard 'benchmark' eljárásokkal számos 'benchmark' gráfon. A kísérleti eredmények igazolták a szerző algoritmusainak hatékonyságát, melyek általában felülmúlták a 'benchmark' eljárásokat különösképpen az ún. 'phase transition' területen ahol az igazán nehéz problémák találhatók. Néhány eredmény a kiterjedt vizsgálatból megtalálható a 8. ábrán, ahol az új Merge Stratégiák különböző Merge Keretrendszerekbe ágyazva kerültek összehasonlításra standard 'benchmark' eljárásokkal. Az ábrákon az algoritmusok jelölései követik a Merge Keretrendszerek UC, CU és CC jelöléseit. Az UC Merge Keretrendszerben az $UC^{choose-col}_{choose-unc}$ jelölést használva $choose - col$ jelöli a színezett sor amely elsőként kerül kiválasztásra, míg a $choose - unc$ jelöli a rákövetkező színezett sor kiválasztási stratégiát. Hasonlóképpen van ez

a CU Merge Keretrendszerben is, csak $CU^{choose-unc}_{choose-col}$ jelöléssel. Továbbá a CC Merge Keretrendszerben a $CC-\text{CHOOSE}$, azonosításban a CHOOSE jelöli az egyetlen sor-pár kiválasztási stratégiát. A kiválasztási függvények a következők szerint jelöltek, a szerző Merge Stratégiái: 'dotprod' - Dot Product; 'cos' - Cosine; '$\tilde{\sigma}$' - közelített Spektrálnorma; 'Zykov$_{\bar{\theta}}$' - Zykov-fa és Lovász-theta stratégia[10] és 'EA^{ζ}' - evolúciós stratégia a ζ-fitnesszel, valamint a kiválasztott 'benchmark' algoritmusok jelölése: 'dsatur' - DSatur heurisztika és 'Erdős' - Erdős heurisztika. Valamint a 'greedy' a mohó választási stratégiát jelöli. Az összehasonlítás egységesítése miatt a 'benchmark' algoritmusok be lettek ágyazva a megfelelő Merge Keretrendszerbe egy megfelelő Merge Modell felhasználásával.

Konklúzió

A szerző új színezési megközelítése a gráfszínezés hatékony modelljének bizonyult. Jelentős csökkentést hozhat az algoritmusok számítási igényében. Továbbá egységes és tömör leírását biztosítja a színezési eljárásoknak, biztosítva ezzel az egységes szerkezetben vett strukturális elemzését. Az algoritmusok implementálása ezen közös módon lehetőséget biztosít az egységes teljesítmény mérésre. Az új színezési keretrendszer általánosítja az eddig színezési sémákat. Ezen általánosítás következményeként egy algoritmus beágyazása a modellbe annak kibővítése mellett jelentős színezési teljesítmény javulással is járhat. Ezen új megközelítés új információ kinyerési technikákat is támogat amely az algoritmus tervezésben segíthet valamint új irányokat adhat a probléma elemzéséhez. A szerző számos hatékony színezési stratégiát és algoritmust mutatott be valamint ismertetett egy általános stratégia készítésre alkalmas módszert amely lehetőséget biztosít mesterséges intelligenciai eljárások alkalmazására a színezési problémához.

[10] A + kitevő jelöli majd a többszöri él hozzáadás alkalmazását.



8. ábra. Az algoritmusok által átlagosan használt színek száma az ún. 'phase transition' területen. A vizsgált gráfok: 10-kromatikusak, egyenlően particionált véletlen gráfok, a generálásnál az élek valószínűsége a következők szerint skálázott: $0.1 \leq p_e \leq 0.9$. Minden p_e élvalószínűséghez 10 db különböző gráf került elkészítésre, $\{1, 2, \ldots, 10\}$ pszeudo véletlen alapbeállítások mellett.

THESIS 1 A szerző a gráfszínezési folyamatot kvóciens és hatványgráfok segítségével, gráf homomorfizmusokon sorozatával definiálta (lásd Juhos et al. [37; 41]). A szerző Kvóciens és Hatvány Módszernek nevezte el az ezen elven alapuló színezési módszereit. A szerző megadta ezen módszereknek konkrét megvalósításait mátrixok és azokon értelmezett speciális műveletek segítségével. Az így keletkező megvalósításokat Merge Modelleknek nevezte el. Ezen modellek alkalmazásával az eredeti probléma transzformálódik és az eredeti gráf homomorf képei jönnek létre. Az új modellek képesek a létező színezési algoritmusok leírására. Egy algoritmus beágyazása a modellbe jelentősen csökkentheti annak számítási igényét. Továbbá a beágyazás az algoritmus egy természetes kiterjesztését is biztosítja, mely annak színezési teljesítményét növelheti. A modell ellentétben a tradicionális sémákkal nem tesz különbséget a színek és a csúcsok között, integrálja őket, mely egy egységes algoritmus szerkezethez vezet ahol a csúcs és szín választás nincs megkülönböztetve.

THESIS 2 A szerző egységesítette és általánosította a szekvenciális színezés korábbi modelljeit, melyeket három Merge Keretrendszerben foglalt össze felhasználva a Merge Modelleket [41; 42]. Ezen keretrendszerek egységes és tömör leírását biztosítják az algoritmusoknak és ezzel az összehasonlításuk egy egységes úton megvalósítható. Továbbá a szerző összefüggéseket vázolt fel az egységes leírásból fakadóan bizonyos algoritmusok színezési teljesítménye és leírása között. A meglévő algoritmusok egy Merge Keretrendszerben való kifejezésével, kihasználva az általánosítást, új algoritmusok formálhatók.

THESIS 3 A szerző bemutatta, hogy egy színezési algoritmus futási teljesítménye javul ha azt egy megfelelő Merge Keretrendszerbe ágyazzuk felhasználva egy megfelelő Merge Modellt. Anélkül, hogy az algoritmus lépéseit megváltoztatnák az új reprezentációnak köszönhetően az algoritmus futási ideje lerövidül [38; 40]. A szerző ezen javulást elméletileg alátámasztotta és kísérletekkel is igazolta. A kísérletek során az ún. 'phase transitions' területen, ahol a problémák nehézzé válnak, vizsgált különböző algoritmusokat, különböző Merge Keretrendszerekben. Továbbá megadott egy módját létező algoritmusok természetes kiterjesztésére egy Merge Keretrendszerben amely a színezési teljesítmény jelentős növelését hozhatja.

THESIS 4 Minden Merge Modellben a színezési művelet az ún. Merge Művelettel lett helyettesítve. Számos Merge Stratégiát fejlesztett ki a szerző, amelyek a Merge Műveletek hatékony sorozatát generálják . A Merge Modellek mátrix reprezentációjának köszönhetően a szerző mátrix normák segítségével is definiált már meglévő heurisztikákat egy alkalmas Merge Keretrendszerbe való ágyazás után, valamint számos általa kifejlesztett stratégiáit is. A szerző által bevezetett stratégiák listája alább olvasható:

– kiterjesztett Hajnal; kiterjesztett Welsh-Powell (∞–norm) [38]

– Spektrálnorma[42]

– közelített Spektrálnorma [42]

– Dot product (entrywise norms) [38]

– Cosine [38]

– Zykov-fa és Lovász-theta [35; 43]

A szerző ezeket a stratégiákat kombinálta különböző Merge Modellekkel és Merge Keretrendszerekkel és számos új algoritmust állított így elő. Ezek teljesítményét összehasonlította számos ismert 'benchmark' algoritmus teljesítményével. Az új algoritmusok többnyire felülmúlták a 'benchmark' algoritmusokat standard gráfokon valamint ezen túlmenően nehezen megoldható problémákon, gráfokon is, az ún. 'phase transition' területen ahol a teszt gráfok alkalmas megválasztása nem torzíthatta a vizsgálat eredményét.

THESIS 5 A szerző bevezette a Merge Útvonal fogalmát [42]. Egy Merge Útvonal a Merge Műveletek során a változó Merge Modell struktúra alapján jön létre. A keletkező mátrixok tulajdonságai alkotják az elemeit. Az útvonal elemei a színezés, a Merge Műveletek lépéseihez kapcsolódnak. A szerző a Merge Műveletek alkalmazási stratégiáit ezen útvonal komponensein keresztül határozta meg. Amely mesterséges intelligenciai eljárások alkalmazását tette lehetővé a színezésben, mint például gépi tanulás és klaszterezés [36] alapján.

THESIS 6 A szerző beágyazta Merge Stratégiáit egy meta-heurisztikus eljárásba, egy evolúciós algoritmusba és létrehozta a következő komponenseket [37–39; 42] :

– Egy mutáció operátort, amely felismeri a nehezen színezhető csúcsokat és előreveszi azokat a színezési folyamatban.

– Egy finomított fitnesz függvényt a hagyományosan alkalmazott fitnesz javítását adja, növelve ezzel az optimalizálás hatékonyságát.

A keletkező meta-heurisztikus algoritmusok jól teljesítettek a 'benchmark' algoritmusokkal való kísérleti összehasonlításban számos 'benchmark' gráfon beleértve az ún. 'phase transition' területet is.

Irodalomjegyzék

[1] Avanthay C, Hertz A, Zufferey N: **A variable neighborhood search for graph coloring**. *European Journal of Operational Research* 2003, **151**:379–388.

[2] Bäck T: *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, New York 1996.

[3] Blochliger I, Zufferey N: **A graph coloring heuristic using partial solutions and a reactive tabu scheme.** *Computers & Operations Research* 2008, **35**:960–975.

[4] Brèlaz D: **New methods to color the vertices of a graph**. *Communications of the ACM* 1979, **22**:251–256.

[5] Bui TN, Patel CM: **An ant system algorithm for coloring graphs**. In *Proceedings of the Computational Symposium on Graph Coloring and its Generalizations* 2002:83–91.

[6] Campelo M, Correa R, Frota Y: **Cliques, holes and the vertex coloring polytope**. *Information Processing Letters* 2004, **89**:159–164.

[7] Caramia M, Dell'Olmo P: **Bounding vertex coloring by truncated multistage branch and bound**. *Networks* 2004, **44**:231–242.

[8] Caramia M, Dell'Olmo P: **Coloring graphs by iterated local search traversing feasible and infeasible solutions**. *Discrete Applied Mathematics* 2008, **156**:201–217.

[9] Chams M, Hertz A, de Werra D: **Some experiments with simulated annealing for coloring graphs**. *European Journal of Operational Research* 1987, **32**:260–266.

[10] Chiarandini M, Dumitrescu I, Stützle T: **Very large-scale neighborhood search: Overview and case studies on coloring problems**. In *European Journal of Operational Research* 2008.

[11] Coll P, Marenco J, Méndez Díaz I, P Z: **Facets of the graph coloring polytope**. *Annals of Operations Research* 2002, **116**:79–90.

[12] Costa D, Hertz A: **Ants Can Colour Graphs**. *Journal of the Operations Research Society* 1997, **48**:295–305.

[13] Costa D, Hertz A, Dubuis C: **Embedding a sequential procedure within an evolutionary algorithm for coloring problems in graphs.** *Journal of Heuristics* 1995, **1**:105–128.

[14] Craenen B, Eiben A, van Hemert J: **Comparing Evolutionary Algorithms on Binary Constraint Satisfaction Problems**. *IEEE Transactions on Evolutionary Computation* 2003, **7**(5):424–444.

[15] Culberson J, Gent I: **Frozen development in graph coloring**. *Theor. Comput. Sci.* 2001, **265**(1–2):227–264.

[16] Culberson J: **Iterated Greedy Graph Coloring and the Difficulty Landscape**. Tech. Rep. TR 92-07 1992.

[17] de Werra D: **Heuristics for Graph Coloring**. *Computational Graph Theory* 1990, **Comput. Suppl. 7**:191–208.

[18] Dukanovic I, Rendl F: **Semidefinite programming relaxations for graph coloring and maximal clique problems**. *Mathematical Programming, Serie B* 2007, **109**:345–365.

[19] Eiben AE, van Der Hauw JK, van Hemert JI: **Graph Coloring with Adaptive Evolutionary Algorithms**. *Journal of Heuristics* 1998, **4**:25–46.

[20] Falkenauer E: *Genetic Algorithms and Grouping Problems*. John Wiley 1998.

[21] Figueiredo R, Barbosa V, Maculan N, de Souza C: **New $0-1$ integer formulations of the graph coloring problem**. In *Proceedings of XI CLAIO* 2002.

[22] Galinier P, Hao JK: **Hybrid evolutionary algorithms for graph coloring**. *Journal of Combinatorial Optimization* 1999, **3**:379–397.

[23] Galinier P, Hertz A: **A survey of local search methods for graph coloring**. *Computers & Operations Research* 2006, **33**:2547–2562.

[24] Gintaras P: **On the Graph Coloring Polytope**. *Information technology and control* 2008, **37**:7–11.

[25] Glass CA, Prügel-Bennett A: **Genetic algorithms for graph colouring: Exploration of Galinier and Hao's algorithm.** *Journal of Combinatorial Optimization* 2003, **7**:229–236.

[26] Graham R, Grötschel M, Lovász L (Eds): *Handbook of Combinatorics*. North-Holland 2005.

[27] Hajnal P: **Eigenvalues of graphs: Graph colouring and the Perron-Frobenius eigenvector**. *Péter Hajnal's Combinatorics seminars. University of Szeged.* 2004.

[28] Halldórsson MM: **A Still Better Performance Guarantee for Approximate Graph Coloring**. *Inf. Process. Lett.* 1993, **45**:19–23.

[29] Hamiez JP, Hao JK: **Scatter search for graph coloring**. In *Artificial Evolution, Volume 2310* 2001:168–179.

[30] Harmanani H, Abas H: **A method for the minimum coloring problem using genetic algorithms**. In *MS'06: Proceedings of the 17th IASTED international conference on Modelling and simulation*, Anaheim, CA, USA: ACTA Press 2006:487–492.

[31] Hertz A, de Werra D: **Using tabu search techniques for graph coloring**. *Computing* 1987, **39**:345–351.

[32] Johnson DS, Mehrotra A, Trick MA: **Special issue on computational methods for graph coloring and its generalizations**. *Discrete Applied Mathematics* 2008, **156**:145–146.

[33] Johnson D, Aragon C, McGeoch L, Schevon C: **Optimization by simulated annealing: an experimental evaluation; part II, graph coloring and number partitioning**. *Operations Research* 1991, **39**:378–406.

[34] Johnson D, Trick M: *Cliques, Coloring, and Satisfiability*. American Mathematical Society, DIMACS 1996.

[35] Juhos I: **Graph Colouring through Clustering**. *Seminar. University of Edinburgh.* 2009.

[36] Juhos I, Szarvas G: **Intelligent Forecast with Dimension Reduction**. In *Applied Soft Computing Technologies: The Challenge of Complexity, Volume 34 of* Advances in Soft Computing. Edited by Abraham A, de Baets B, Köppen M, Nickolay B, Springer 2006:279–292.

[37] Juhos I, Tóth A, van Hemert J: **Binary Merge Model Representation of the Graph Colouring Problem**. In *Evolutionary Computation in Combinatorial Optimization, Volume 3004 of* Lecture Notes in Computer Science. Edited by Gottlieb J, Raidl, Raidl GR, Springer 2004:124–134.

[38] Juhos I, Tóth A, van Hemert J: **Heuristic Colour Assignment Strategies for Merge Models in Graph Colouring**. In *Evolutionary Computation in Combinatorial Optimization, Volume 3448 of* Lecture Notes in Computer Science. Edited by Gottlieb J, Raidl, Raidl GR, Springer 2005:132–143.

[39] Juhos I, van Hemert J: **Improving graph colouring algorithms and heuristics using a novel representation**. In *Evolutionary Computation in Combinatorial Optimization, Volume 3906 of* Lecture Notes in Computer Science. Edited by Gottlieb J, Raidl, Raidl GR, Springer 2006:123–134.

[40] Juhos I, van Hemert J: **Increasing the efficiency of graph colouring algorithms with a representation based on vector operations.** *Journal of Software* 2006, **1**(2):24–33.

[41] Juhos I, van Hemert J: *Contraction-based heuristics to improve the efficiency of algorithms solving the graph colouring problem*, Springer, *Volume 153 of* Studies in Computational Intelligence 2008 chap. III, :175–192.

[42] Juhos I, van Hemert J: **Graph Colouring Heuristics Guided by Higher Order Graph Properties**. In *Evolutionary Computation in Combinatorial Optimization, Volume 4972 of* Lecture Notes in Computer Science. Edited by van Hemert J, Cotta C, Springer 2008:97–109.

[43] Juhos I, van Hemert JI: **Graph colouring through clustering by Zykov-tree and Lovász-theta**. *Submitted* 2009.

[44] Karger D, Motwani R, Sudan M: **Approximate graph coloring by semidefinite programming**. *Journal of the ACM* 1998, **45**(2):246–265.

[45] Karp RM: *Complexity of Computer Computations: Reducibility Among Combinatorial Problems*, New York: Plenum 1972 :85–103.

[46] Leighton F: **A graph coloring algorithm for large scheduling problems**. *Journal of Reasearch of the National Bureau of Standards* 1979, **84**:489–506.

[47] Malaguti E, Toth P: **A survey on vertex coloring problems**. *International Transactions in Operational Research* 2009, :1–34.

[48] Marino A, Damper R: **Breaking the Symmetry of the Graph Colouring Problem with Genetic Algorithms**. In *Late Breaking Papers at the 2000 Genetic and Evolutionary Computation Conference*. Edited by Whitley D 2000:240–245.

[49] Matula D, Marble G, J I: *Graph coloring algorithms*, Academic Press 1972 :109–122.

[50] Mehrotra A, Trick MA: **A column generation approach for graph coloring**. INFORMS *Journal on Computing* 1996, **8**:344–354.

[51] Méndez-Díaz I, Zabala P: **A polyhedral approach for graph coloring**. *Electronic Notes in Discrete Mathematics* 2001, **7**:178–181.

[52] Méndez-Díaz IP Zabala: **A cutting plane algorithm for graph coloring**. *Discrete Applied Mathematics* 2008, **156**:159–179.

[53] Méndez-Díaz I, Zabala P: **A branch-and-cut algorithm for graph coloring**. *Discrete Appl. Math.* 2006, **154**(5):826–847.

[54] Merikoski JK, Kumar R: **Lower Bounds for the Spectral Norm**. *Journal of Inequalities in Pure and Applied Mathematics* 2005, **6**(3).

[55] Palubeckis G: **On the recursive largest first algorithm for graph colouring**. *Int. J. Comput. Math.* 2008, **85**(2):191–200.

[56] Porumbel D, Hao JK, Kuntz P: **Diversity control and multi-parent recombination for evolutionary graph coloring algorithms**. In *LNCS 5482*, Springer 2009.

[57] Schindl D: **Graph coloring and linear programming**. In *LNCS 5482*, Presentation at First Joint Operations Research Days, Ecole Polytechnique Fédérale de Lausanne (EPFL), Available on line 2003.

[58] Sewell E: **An improved algorithm for exact graph coloring**. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* 2006, **26**:359–373.

[59] van Hemert J: **Application of Evolutionary Computation to Constraint Satisfaction and Data Mining**. *PhD thesis*, Leiden University 2002.

[60] Welsh DJA, Powell MB: **An upper bound for the chromatic number of a graph and its application to timetabling problems**. *The Computer Journal* 1967, **10**:85–86.

[61] Wigderson A: **Improving the performance for approximate graph coloring**. *Journal of the ACM* 1983, **30**:729–735.

[62] Wilf HS: **The eigenvalues of a graph and its chromatic number**. *Journal of London Math. Soc.* 1967, :330–332.

[63] Woeginger G: **Exact Algorithms for NP-Hard Problems: A Survey**. In *Journal of London Math. Soc.* 2003:185–207.

[64] Zykov AA: **On some properties of linear complexes (in Russian)**. *Math. Sbornik* 1949, **24**:163–188.

[65] Zykov AA: **On some properties of linear complexes**. *American Mathematical Society Translations* 1952, **79**:81.