

HRZZ istraživački projekt IP-2020-02-9752
Algoritamske konstrukcije kombinatornih objekata

GAP paket za konstrukciju kombinatornih objekata

Interni izvještaj

Vedran Krčadinac, Anamari Nakić,
Mario Osvin Pavčević, Renata Vlahović Kruc

Verzija 6, 14.12.2021.

Sadržaj

1	HRZZ projekt ACCO	1
2	Postojeći GAP paketi	2
2.1	GRAPE	2
2.2	DESIGN	3
2.3	FinInG	3
2.4	GUAVA	3
2.5	Digraphs	4
2.6	images	4
2.7	AGT	5
2.8	AssociationSchemes	5
2.9	COCO2P	5
2.10	UnitalSZ	5
2.11	NautyTracesInterface	6
2.12	BlissInterface	6
3	Funkcionalnost našeg paketa	6
4	Strukture podataka	7
5	Povezivanje s vanjskim programima	9
6	Vremenski plan razvoja novog paketa	9
7	Literatura	10

1 HRZZ projekt ACCO

Tema istraživačkog projekta *Algoritamske konstrukcije kombinatornih objekata* (ACCO) su kombinatorni objekti s dodatnom algebarskom strukturom, kao što su kvazisimetrični dizajni, shematski dizajni, q -analogoni dizajna, diferencijski skupovi, (semi)parcijalne geometrije i njihove generalizacije. Rezultati iz algebarske kombinatorike nameću ograničenja za dopustive parametre i svojstva takvih objekata koja se mogu iskoristiti da bi se suzio prostor pretraživanja i razvili specijalizirani algoritmi za njihovu konstrukciju i klasifikaciju. Osnovni ciljevi projekta su:

- O1. Razvoj algoritamskih metoda konstrukcije i klasifikacije kombinatornih objekata s dodatnom algebarskom strukturom. Razvijene metode koristit će algebarska i kombinatorna svojstva objekata da bi bile primjenjive na objekte s većim parametrima i probleme koji su izvan dosega poznatih konstrukcijskih metoda.
- O2. Proširivanje teorijskog znanja o kombinatornim objektima koji su predmet istraživanja. Zanimljivi teoremi često se otkrivaju i dokazuju na temelju dostupnih primjera. Očekujemo da će rezultati ovog projekta dovesti do takvih otkrića.
- O3. Razvoj softverskog paketa, implementiranog u sustavu GAP [3], za konstrukciju i analizu kombinatornih objekata.

Ovaj izvještaj je korak prema trećem cilju (O3). Softverski alati koje smo koristili u prethodnim istraživanjima [1, 7, 8, 9, 10, 11, 12, 13, 14, 18] su sustav za računalnu algebru GAP [3], programi za ispitivanje izomorfности i računanje pune grupe automorfizama *nauty* i *Traces* B. D. McKaya i A. Piperna [15], solver A. Wassermana [19] zasnovan na LLL algoritmu, program *cliquer* P. R. J. Östergarda i S. Niskanena [16, 17] za traženje klika u grafovima te naši vlastiti programi pisani u programskom jeziku C za druge probleme koji zahtijevaju puno računalnog vremena. Omogućili smo pozivanje tih programa iz sustava GAP koji koristimo kao sučelje za konstruiranje kombinatornih objekata. Ovaj softverski paket je u ranoj fazi razvoja, bez dokumentacije i sustavnog pristupa strukturama podataka za reprezentiranje objekata i međusobnom povezivanju dijelova sustava. Smisao cilja O3 je otkloniti navedene nedostatke, proširiti sustav i objaviti ga u obliku GAP paketa.

Postoji određen broj prihvaćenih i distribuiranih GAP paketa za rad s kombinatornim objektima (GRAPE, DESIGN, FinInG,...). U ovom izvještaju analiziramo postojeća rješenja i naš ranije razvijeni kod, opisujemo planiranu funkcionalnost novog paketa te specificiramo strukture podataka i protokole za povezivanje s vanjskim programima i postojećim GAP paketima.

2 Postojeći GAP paketi

GAP – *Groups, Algorithms, and Programming* [3] je sustav za računalnu algebru s naglaskom na računalnu teoriju grupa. Aktualno izdanje GAP 4.11.1 dostupno je na adresi <https://www.gap-system.org>. GAP uključuje programski jezik, biblioteku funkcija pisanih u GAP jeziku koje implementiraju razne algebarske algoritme te bazu podataka algebarskih objekata. Biblioteke programa i funkcija koje razvijaju korisnici nazivaju se GAP paketima. Neki od paketa distribuiraju se zajedno s osnovnim GAP sustavom, ali autori paketa su odgovorni za njihovo održavanje i razvoj. Popis svih GAP paketa dostupan je na <http://gap-packages.github.io>.

U nastavku dajemo pregled postojećih GAP paketa za rad s kombinatornim objektima i kratko opisujemo njihov značaj za razvoj našeg paketa.

2.1 GRAPE

Puni naziv: *GRAPE – GRaph Algorithms using PERmutation groups*.

Autor: Leonard H. Soicher.

Zadnja dostupna verzija: 4.8.5 (2021.).

GRAPE je paket za računanje s grafovima i grupama, razvijen primarno za konstruiranje i analiziranje grafova povezanih s konačnim geometrijama i dizajnama. Uključuje funkcije za konstruiranje i modificiranje grafova, ispitivanje svojstava grafova i njihove regularnosti, određivanje nekih posebnih podskupova skupa vrhova i bojanja grafova te izračunavanje punih grupa automorfizama i testiranje izomorfnosti grafova. Mnogi drugi GAP paketi koriste funkcije iz ovog paketa. U našem paketu koristit ćemo strukturu podataka za grafove iz paketa GRAPE i neke od funkcija. Također, analizirat ćemo način na koji GRAPE komunicira s vanjskim programima i na sličan način implementirati povezivanje našeg paketa s programima pisanim u programskom jeziku C.

2.2 DESIGN

Puni naziv: *DESIGN*.

Autor: Leonard H. Soicher.

Zadnja dostupna verzija: 1.7 (2019.).

DESIGN je paket za računanje s blokovnim dizajnima. Uključuje funkcije za konstruiranje dizajna, određivanje njihovih parametara i drugih svojstava, izračunavanje punih grupa automorfizama i testiranje izomorfности dizajna, klasificiranje malih blokovnih dizajna i latinskih kvadrata te particioniranje blokova dizajna. U paketu DESIGN implementirana je XML struktura podataka za dizajne razvijena u sklopu projekta “A web-based resource for design theory” [2]. U našem paketu dizajne i druge incidencijske strukture pohranjivat ćemo koristeći se tom strukturom podataka.

2.3 FinInG

Puni naziv: *FinInG – Finite Incidence Geometry*.

Autori: John Bamberg, Anton Betten, Philippe Cara, Jan De Beule, Michel Lavrauw i Max Neunhoeffler.

Zadnja dostupna verzija: 1.4.1 (2018.).

FinInG je paket za računanje s konačnim incidencijskim geometrijama. Uključuje funkcije za rad s projektivnim prostorima, projektivnim grupama, polaritetima projektivnih prostora, klasičnim polarnim prostorima, afinim prostorima, algebarskim mnogostrukostima nad konačnim poljima, generaliziranim poligonima i dijagramskim geometrijama. Za projekt ACCO od interesa su funkcije iz ovog paketa koje možemo koristiti za konstruiranje q -analogona dizajna, (semi)parcijalnih geometrija i njihovih generalizacija.

2.4 GUAVA

Puni naziv: *GUAVA*.

Autori: Reinald Baart, Tom Boothby, Jasper Cramwinckel, Joe Fields, David Joyner, Robert Miller, Eric Minkes, Erik Roijackers, Lea Ruscio i Cen Tjhai.

Zadnja dostupna verzija: 3.15 (2019.).

GUAVA je paket za računanje s kodovima. Uključuje funkcije za kons-

truiranje i modificiranje nelinearnih, linearnih i cikličkih kodova, kodiranje/dekodiranje i ispravljanje pogrešaka te analiziranje svojstava kodova kao što su minimalna udaljenost i distribucija težina. U dosadašnjim istraživanjima koristili smo paket GUAVA za analiziranje kodova pridruženih dizajnim. Neke od funkcija iz ovog paketa su manje efikasne od odgovarajućih funkcija iz drugog sustava za računalnu algebru MAGMA. U slučajevima kad je računanje trajalo predugo, prebacivali smo dizajne i kodove iz GAP formata u MAGMA format da bismo ih tamo analizirali. U sklopu planiranog GAP paketa namjeravamo automatizirati prebacivanje konstruiranih objekata u format prikladan za sustav MAGMA.

2.5 Digraphs

Puni naziv: *Digraphs – Graphs, digraphs, and multidigraphs in GAP.*

Autori: Jan De Beule, Julius Jonušas, James Mitchell, Michael Torpey, Maria Tsalakou i Wilf A. Wilson.

Zadnja dostupna verzija: 1.3.1 (2020.).

Digraphs je paket za računanje s usmjerenim grafovima i njihovim generalizacijama. Funkcionalnost je slična kao u paketu GRAPE, ali koristi druge strukture podataka za (usmjerene) grafove. Sadrži i funkcije za prebacivanje u GRAPE format. Paket Digraphs i neke od njegovih funkcija pozivaju se iz paketa FinInG. Moguće je da ćemo ih koristiti u našem paketu.

2.6 images

Puni naziv: *images – Minimal and canonical images.*

Autori: Christopher Jefferson, Markus Pfeiffer, Rebecca Waldecker i Eliza JonauskYTE.

Zadnja dostupna verzija: 1.3.1 (2021.).

Paket images implementira efikasne algoritme za izračunavanje minimalnih i kanonskih slika nekoliko vrsta objekata pod djelovanjem permutacijske grupe, uključujući podskupove i familije podskupova. Proučit ćemo funkcije i metode iz ovog paketa i usporediti ih s našim algoritmima za generiranje predstavnika orbita podskupova.

2.7 AGT

Puni naziv: *AGT – Algebraic Graph Theory package.*

Autor: Rhys J. Evans.

Zadnja dostupna verzija: 0.2 (2020.).

Paket AGT proširuje paket GRAPE funkcijama za ispitivanje kombinatoričkih i algebarskih svojstava grafova. Sadrži biblioteku jako regularnih grafova do 40 vrhova i konstrukcije nekoliko familija grafova. Moguće je da ćemo koristiti neke funkcije iz paketa AGT u našem paketu.

2.8 AssociationSchemes

Puni naziv: *AssociationSchemes.*

Autori: John Bamberg, Akihide Hanaki i Jesse Lansdown.

Zadnja dostupna verzija: 1.0.0 (2019.).

Ovo je GAP paket za rad s asocijacijskim shemama i homogenim koherentnim konfiguracijama. Uključuje funkcije za konstruiranje, modificiranje i analiziranje ovih objekata te odgovarajućih algebra i pravilnih podstruktura. Koristit ćemo ga u našem paketu za analiziranje asocijacijskih shema pridruženih shematskim dizajnima.

2.9 COCO2P

Puni naziv: *COCO2P.*

Autori: Mikhail Klin, Christian Pech i Sven Reichard.

Zadnja dostupna verzija: 0.19 (2020.).

Ovo je drugi GAP paket za rad s koherentnim konfiguracijama, implementiranim kao grafovi s obojanim bridovima. Proučit ćemo funkcionalnost paketa COCO2P i usporediti ga s paketom AssociationSchemes.

2.10 UnitalSZ

Puni naziv: *UnitalSZ – Algorithms and library of abstract unitals and their embeddings.*

Autori: Gábor P. Nagy i Dávid Mezőfi.

Zadnja dostupna verzija: 0.16 (2020.).

UnitalSZ je paket za rad s apstraktnim unitalima i unitalima uloženim u projektivne ravnine. Uključuje biblioteku malih primjera koja sadrži i unitale konstruirane u našim radovima [7, 11]. Moguće je da ćemo neke od funkcija iz paketa UnitalSZ koristiti u našem paketu.

2.11 NautyTracesInterface

Puni naziv: *NautyTracesInterface*.

Autor: Sebastian Gutsche.

Zadnja dostupna verzija: 0.2 (2018.).

Paket koji povezuje GAP s programima *nauty* i *Traces* B. D. McKaya i A. Piperna [15]. Proučit ćemo metodu povezivanja iz ovog paketa i usporediti je s metodom iz paketa GRAPE.

2.12 BlissInterface

Puni naziv: *BlissInterface*.

Autor: Gábor P. Nagy.

Zadnja dostupna verzija: 0.22 (2020.).

Paket koji povezuje GAP s programom *bliss* T. Junttila i P. Kaskija [5]. Proučit ćemo metodu povezivanja iz ovog paketa i usporediti je s drugim paketima.

3 Funkcionalnost našeg paketa

Glavna funkcionalnost planiranog GAP paketa je konstruiranje konačnih objekata sa zadanim grupama automorfizama. Osnovne metode konstrukcije [4, 6] intenzivno se koriste od 1980-tih godina za blokovne dizajne i slične strukture. U radovima [1, 11, 13] prilagodili smo ih nekim kombinatornim objektima koje proučavamo u sklopu projekta ACCO. GAP sadrži opsežnu biblioteku funkcija za računanje s permutacijskim grupama, s pomoću kojih zadajemo grupu G i njezino djelovanje na skup S . U paketu ćemo implementirati korake osnovnih metoda konstrukcije objekata s grupom automorfizama G , algoritme prilagođene kombinatornim objektima s dodatnom algebarskom strukturom i pomoćne algoritme za analiziranje konstruiranih objekata. Planirana funkcionalnost paketa uključuje:

- Brzo generiranje predstavnika orbita k -članih podskupova od S pod djelovanjem grupe G ; algoritam je razvijen u [13].
- Generiranje predstavnika orbita k -članih podskupova sa zadanim veličinama presjeka; algoritam je razvijen i korišten u [9, 14].
- Generiranje matrica taktičke dekompozicije (orbitnih matrica) za t -dizajne; algoritmi su razvijeni i korišteni u [7, 11, 12].
- Generiranje predstavnika orbita k -članih podskupova usklađenih sa zadanom orbitnom matricom; razvijeno i korišteno u [11, 14].
- Generiranje Kramer-Mesnerove matrice od zadanog skupa predstavnika orbita t -članih i k -članih podskupova; korišteno u [11] i u drugim radovima.
- Rješavanje Kramer-Mesnerovog sustava s pomoću solvera A. Wassermana [19], vlastitih solvera iz [11, 13] te specijaliziranih solvera koje ćemo razviti za konkretne probleme.
- Sastavljanje traženih objekata iz predstavnika orbita na temelju traženja klika u grafovima. U radovima [1, 10, 14] koristili smo program *cliquer* P. R. J. Östergarda i S. Niskanena [16, 17]. Planiramo upotrijebiti i druge programe za traženje klika.
- Ispitivanje izomorfности konstruiranih objekata i računanje pune grupe automorfizama s pomoću programa *nauty* i *Traces* B. D. McKaya i A. Piperna [15] te programa *bliss* T. Junttila i P. Kaskija [5].
- Određivanje veličina presjeka blokova zadanog dizajna, izračunavanje blokovnog grafa pridruženog kvazisimetričnom dizajnu ili konfiguraciji, izračunavanje asocijacijske sheme pridružene shematskom dizajnu i druge funkcije za analiziranje konstruiranih objekata. Neke od tih funkcija razvit ćemo u GAP-ovom programskom jeziku. Za probleme koji zahtijevaju puno računalnog vremena, algoritme ćemo implementirati u programskom jeziku C i povezati ih s GAP paketom.

4 Strukture podataka

U dosadašnjim istraživanjima konstruirane objekte pohranjivali smo na sljedeće načine. Unutar sustava GAP:

- Kao liste podskupova, pri čemu su podskupovi također liste koje sadrže elemente osnovnog skupa (točke). Na taj način smo reprezentirali dizajne, grafove i druge incidencijske strukture.
- Kao incidencijske matrice, koje se u GAP-u implementiraju kao liste redaka. Recipročne su liste koje sadrže nule i jedinice.

Izvan sustava GAP:

- Kao tekstualne datoteke koje sadrže incidencijske matrice. Na početku datoteke je broj redaka i stupaca, a nakon toga slijedi niz znakova 0 i 1. Recipročne matrice označeni su znakom prelaska u novi red.
- Kao tekstualne datoteke u kojima su pohranjene liste podskupova. Ovo je u osnovi GAP format spremljen u tekstualnu datoteku.
- Za grafove: tekstualne datoteke koje počinju brojem vrhova, a zatim slijedi po jedan brid u svakom retku (dva vrha razdvojena prazninom). Kraj niza bridova označen je brojem -1 .

U planiranom paketu namjeravamo standardizirati strukture podataka koje koristimo u sustavu GAP. Za pohranu grafova koristit ćemo strukturu podataka iz paketa GRAPE. Graf se pohranjuje kao zapis (*record*) s obveznim komponentama `isGraph`, `order`, `group`, `schreierVector`, `representatives` i `adjacencies`. Za kreiranje grafova koriste se GRAPE funkcije koje automatski postavljaju vrijednosti odgovarajućih zapisa. Za dizajne i druge incidencijske strukture koristit ćemo strukturu podataka iz paketa DESIGN. Dizajn D pohranjuje se kao zapis s obveznim komponentama `isBlockDesign`, `v` i `blocks`. Točke dizajna su uvijek brojevi $1, 2, \dots, D.v$, ali moguće im je dati imena u neobveznoj komponenti `pointNames`. U tom slučaju `D.pointNames[i]` je ime točke i . Komponenta `blocks` je uzlazno sortirana lista blokova od D , s mogućim ponavljanjem blokova. Svaki blok je uzlazno sortirana lista točaka. Druge neobvezne komponente zapisa u kojem je pohranjen dizajn su `isSimple`, `isBinary`, `isConnected`, `r`, `blockSizes`, `blockNumbers`, `resolutions`, `autGroup`, `autSubgroup`, `tSubsetStructure`, `allTDesignLambdas`, `efficiency`, `id`, `statistical_propertiesXML`.

Paket će uključivati funkcije za učitavanje i pohranjivanje objekata u tekstualne datoteke u formatima koje smo ranije koristili. Također, omogućit ćemo pohranjivanje u tekstualni format prikladan za sustav MAGMA.

5 Povezivanje s vanjskim programima

U dosadašnjem radu sa sustavom GAP, povezivali smo ga s vanjskim programima pisanjem i čitanjem tekstualnih datoteka. Isti pristup koristi se u paketu GRAPE, ali je tehnički bolje riješen. Jedan od problema ovog pristupa su konflikti s imenima datoteka kad se paralelno izvodi više kopija programa GAP. To smo ranije rješavali tako da smo imenima datoteka dodavali broj koji se povećava sa svakim kreiranjem nove datoteke. Sve pomoćne datoteke spremali smo u isti direktorij u kojem je također pohranjen broj koji dodajemo imenima datoteka. U rijetkim situacijama ipak je dolazilo do konflikta, ako više paralelnih GAP procesa istovremeno čita ili piše datoteku u kojoj je pohranjen broj.

U paketu GRAPE svaka kopija programa GAP piše i čita pomoćne datoteke u drugi privremeni direktorij. Koristi se GAP funkcija `Directory-Temporary()` koja kreira jedinstveni privremeni direktorij. Prije nego završi, GAP proces briše sve privremene direktorije koje je kreirao. Na taj način ne dolazi do konflikta s imenima datoteka i moguće je izvođenje većeg broja paralelnih GAP procesa. Na sličan način riješit ćemo povezivanje s vanjskim programima u našem novom GAP paketu za konstrukciju kombinatornih objekata.

Drugi GAP paketi koriste drugačije pristupe za komunikaciju s vanjskim programima. Proučit ćemo pakete `NautyTracesInterface` i `BlissInterface` i usporediti ih s rješenjem iz paketa GRAPE. Za sada je plan koristiti rješenje s pisanjem i čitanjem tekstualnih datoteka jer je najbližnije našim dosadašnjim GAP programima. Ako se pokaže da su druga rješenja efikasnija ili jednostavnija, možda se odlučimo za neki drugi način povezivanja s vanjskim programima.

6 Vremenski plan razvoja novog paketa

U prvom projektnom razdoblju koristili smo se s ranije razvijenim GAP programima. Oni nisu uobličeni kao formalni GAP paket, nego kao datoteka koja sadrži GAP kod i učita se na početku rada. Proučili smo srodne GAP pakete i upoznali se s njihovim strukturama podataka i rješenjima za povezivanje s vanjskim programima.

U drugom projektnom razdoblju kreirat ćemo prvu verziju novog paketa po specifikacijama za GAP pakete. Prepraviti ćemo ranije razvijene programe

tako da zadovoljavaju zahtjeve za GAP pakete, rade sa strukturama podataka opisanim u cjelini 4 i povezuju se s vanjskim programima kao što je opisano u cjelini 5. Paralelno ćemo pisati dokumentaciju za implementirane metode.

U razvoju, testiranju i korištenju prve verzije paketa sudjelovat će svi članovi projekta. Na taj način ćemo detektirati i ispravljati pogreške u kodu i usmjeravati razvoj novih metoda za konstrukciju i analizu kombinatornih objekata. Prije kraja trećeg projektnog razdoblja očekujemo da ćemo imati drugu verziju paketa, koju ćemo objaviti kao vanjski doprinos GAP-u.

7 Literatura

- [1] M. Abreu, M. Funk, V. Krčadinac, D. Labbate, *Strongly regular configurations*, preprint, 2021.
- [2] R. A. Bailey, P. J. Cameron, P. Dobcsányi, J. P. Morgan, L. H. Soicher, *Designs on the Web*, Discrete Math. **306** (2006), 3014–3027.
- [3] The GAP Group, *GAP – Groups, Algorithms, and Programming*, Version 4.11.1, 2021. <https://www.gap-system.org>
- [4] Z. Janko, T. van Trung, *Construction of two symmetric block designs for (71, 21, 6)*, Discrete Math. **55** (1985), no. 3, 327–328.
- [5] T. Junttila, P. Kaski, *bliss: A Tool for Computing Automorphism Groups and Canonical Labelings of Graphs*, 2015. <http://www.tcs.hut.fi/Software/bliss/>
- [6] E. S. Kramer, D. M. Mesner, *t-designs on hypergraphs*, Discrete Math. **15** (1976), no. 3, 263–296.
- [7] V. Krčadinac, *Steiner 2-designs $S(2, 4, 28)$ with nontrivial automorphisms*, Glas. Mat. Ser. III **37(57)** (2002), no. 2, 259–268.
- [8] V. Krčadinac, *Some new designs with prescribed automorphism groups*, J. Combin. Des. **26** (2018), 193–200.
- [9] V. Krčadinac, *Non-embeddable quasi-residual quasi-symmetric designs*, Appl. Algebra Engrg. Comm. Comput. (2020), 7 str.

- [10] V. Krčadinac, *A new partial geometry $pg(5,5,2)$* , J. Combin. Theory Ser. A **183** (2021), 105493.
- [11] V. Krčadinac, A. Nakić, M. O. Pavčević, *The Kramer-Mesner method with tactical decompositions: some new unitals on 65 points*, J. Combin. Des. **19** (2011), no. 4, 290–303.
- [12] V. Krčadinac, A. Nakić, M. O. Pavčević, *Equations for coefficients of tactical decomposition matrices for t -designs*, Des. Codes Cryptogr. **72** (2014), no. 2, 465–469.
- [13] V. Krčadinac, R. Vlahović, *New quasi-symmetric designs by the Kramer-Mesner method*, Discrete Math. **339** (2016), no. 12, 2884–2890.
- [14] V. Krčadinac, R. Vlahović Kruc, *Quasi-symmetric designs on 56 points*, Adv. Math. Commun. (2020), 14 str.
- [15] B. D. McKay, A. Piperno, *Practical Graph Isomorphism, II*, J. Symbolic Comput. **60** (2014), 94–112.
- [16] S. Niskanen, P. R. J. Östergard, *Cliquer user's guide*, version 1.0, Communications Laboratory, Helsinki University of Technology, Espoo, Finland, Tech. Rep. T48, 2003.
- [17] P. R. J. Östergard, *A fast algorithm for the maximum clique problem*, Discrete Appl. Math. **120** (2002), no. 1–3, 197–207.
- [18] R. Vlahović Kruc, *Neki rezultati o kvazisimetričnim dizajnima s iznimnim parametrima*, doktorska disertacija, Sveučilište u Zagrebu, 2019.
- [19] A. Wassermann, *Finding simple t -designs with enumeration techniques*, J. Combin. Des. **6** (1998), no. 2, 79–90.