



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MATEMATIKY

INSTITUTE OF MATHEMATICS

**URYCHLENÍ VÝPOČTU GRAMATICKÉ EVOLUCE
POMOCÍ METODY "KERNEL TRICK"**

ACCELERATION OF GRAMMATICAL EVOLUTION CALCULATION USING THE "KERNEL TRICK" METHOD

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Anna Kučerová

VEDOUCÍ PRÁCE

SUPERVISOR

prof. Ing. Pavel Ošmera, CSc.

BRNO 2020

Zadání bakalářské práce

Ústav: Ústav matematiky
Studentka: **Anna Kučerová**
Studijní program: Aplikované vědy v inženýrství
Studijní obor: Matematické inženýrství
Vedoucí práce: **prof. Ing. Pavel Ošmera, CSc.**
Akademický rok: 2019/20

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Urychlení výpočtu gramatické evoluce pomocí metody "Kernel trick"

Stručná charakteristika problematiky úkolu:

Pro optimalizaci složitých systémů se používá gramatická evoluce. Pro testovací úlohy je dostupná verze PonyGE2 (v jazyce Python). Nevýhodou je poměrně dlouhá doba výpočtu. Nedávno byla publikovaná metoda "Kernel trick", která používá statistické metody pro nalezení vhodných jedinců prohledávaného prostoru, které jsou blízko cílové funkce (fitness).

Cíle bakalářské práce:

Student nastuduje jádrovou metodu a evoluční algoritmy se zaměřením na gramatickou evoluci. Popíše software PonyGE2 pro přípravu na implementaci jádrových metod.

Seznam doporučené literatury:

LANE, Fergal, AZAD, Muhammad Atif R. and RYAN, Conor. Principled Evolutionary Algorithm Design and the Kernel Trick. Genetic and Evolutionary Computation Conference (GECCO). USA, 2016.

RYAN, Conor. Gecco 2006 Grammatical Evolution Tutorial. USA, 2006.

ORTEGA, Alfonso, CRUZ, Marina and ALFONSECA, Manuel. Christiansen Grammar Evolution: Grammatical Evolution With Semantics. IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Journal. USA, 2007.

O'NEILL, Michael, HEMBERG, Erik, GILLIGAN, Conor, BARTLEY, Elliott, MCDERMOTT, James a BRABAZON, Anthony. GEVA : grammatical evolution in Java. ACM SIGEVOLUTION Journal. New York, USA, 2008.

MIŠKARÍK, Kamil. GRAMATICKÁ EVOLUCE - JAVA MATLAB IMPLEMENTACE. VUT Brno. Brno, ČR, 2013.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2019/20

V Brně, dne

L. S.

prof. RNDr. Josef Šlapal, CSc.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

Abstrakt

Tato práce se zabývá popisem evolučních algoritmů, konkrétně pak gramatické evoluce, která je implementována do optimalizačního programu PonyGE2. Dále popisuje princip jádrové metody.

Abstract

This thesis describing evolutionary algorithms, specifically grammatical evolution, which is implementing in optimization software PonyGE2. Further describes the principle of the kernel trick.

Klíčová slova

Evoluční algoritmy, gramatická evoluce, PonyGE2, jádrová metoda.

Keywords

Evolutionary algorithms, grammatical evolution, PomyGE2, kernel trick.

KUČEROVÁ, Anna. *Urychlení výpočtu gramatické evoluce pomocí metody "Kernel trick"* [online]. Brno, 2020 [cit. 2020-06-25]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/125402>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav matematiky. Vedoucí práce Pavel Ošmera.

Prohlašuji, že jsem bakalářskou práci *Urychlení výpočtu gramatické evoluce pomocí metody "Kernel trick"*, vypracovala sama pod vedením prof. Ing. Pavla Ošmery CSc. s použitím materiálů uvedených v seznamu literatury.

Anna Kučerová

Na tomto místě bych ráda poděkovala Ing. Rudolfu Müllerovi a Ing. Františku Zemanci za konzultace ohledně dílčích částí bakalářské práce a mému vedoucímu bakalářské práce prof. Ing. Pavlu Ošmerovi CSc. za připomínky a korekce bakalářské práce.

Anna Kučerová

Úvod	12
1. Evoluční algoritmy	13
1.1. Genetické algoritmy	14
1.2. Pojmy	15
1.1. Variační operátory	16
2. Gramatická evoluce	17
2.1. Pojmy	17
2.2. Backus-Naurova forma	18
3. Popis algoritmu PonyGE2.....	19
3.1. Gramatika.....	19
3.2. Chromozomová reprezentace, derivační strom a inicializace	19
3.3. Křížení	21
3.4. Mutace.....	22
3.5. Selektce.....	22
3.6. Záměna populace	22
3.8. Vyhodnocení populace.....	23
4. Testovací úloha algoritmu PonyGE2	24
4.1. Statistika výsledků.....	26
5. Jádrové metody	29
5.1. Jádrová funkce.....	29
5.2. Základní jádrové operace a mutace	30
5.3. Lineární kombinace, hyperplochy a bázové prostory	31
5.4. Souřadnicový systém.....	32
5.5. Křížení	33
Závěr.....	34
Reference	35
Přílohy.....	36

Úvod

Zadáním bakalářské práce je seznámit se s evolučními algoritmy, které se využívají v optimalizacích a díky své variabilitě získávají v posledních letech na popularitě mezi optimalizačními metodami.

Další kapitola se bude věnovat gramatické evoluci, což je skupina algoritmů spadající mezi evoluční algoritmy. Motivací pro popis těchto algoritmů je následná práce s optimalizačním softwarem PonyGE2, který gramatickou evoluci využívá, a kterému se budou věnovat hned dvě následující části.

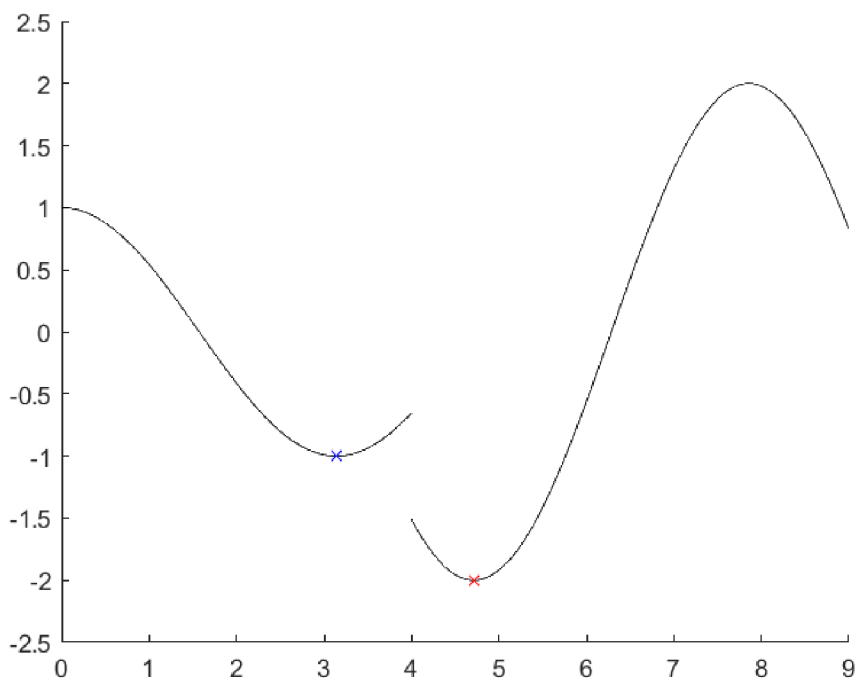
Nejprve bude optimalizační algoritmus PonyGE2 popsán (složení softwaru) a následně využit pro navrženou testovací úlohu. Ta bude volena tak, aby ukázala principy chování algoritmu (tedy optimalizací pomocí gramatické evoluce).

Poslední kapitola se bude týkat jádrových metod. Tyto algoritmy se ve strojovém učení využívají pro analýzu vzorů. Jádrové metody mají velký potenciál v urychlení výpočtu optimalizací, proto se opět zaměříme na úvod do problematiky.

1. Evoluční algoritmy

Evoluční algoritmy jsou souborem algoritmů vycházejících z biologických poznatků adaptace v přírodě, které stojí na simulaci dědičnosti a evolučních problémů. Běžně se využívají pro řešení obtížných výpočetních úloh s nelineární složitostí. Většina evolučních algoritmů spadá do skupiny stochastických algoritmů, obvykle kombinovanými s metodami přímého prohledávání a běžně se využívají pro řešení optimalizačních úloh. Inspirace v živé přírodě vede k vytváření vhodných postupů pro řešení daného problému. Neexistuje však obecně platný postup pro libovolnou úlohu.

Výhodou těchto algoritmů je řešení obtížně optimalizovatelných funkcí. Mezi takovéto funkce spadají funkce definované pouze numericky nebo nespojitě funkce obsahující skokové změny, viz graf 1.a. Další významnou výhodou, která plyne z vlastností stochastických algoritmů je, možnost překonání lokálního maxima (resp. minima) a konvergence ke globálnímu maximu (resp. minimu). Tím se sníží riziko uvíznutí a předčasného ukončení optimalizace.



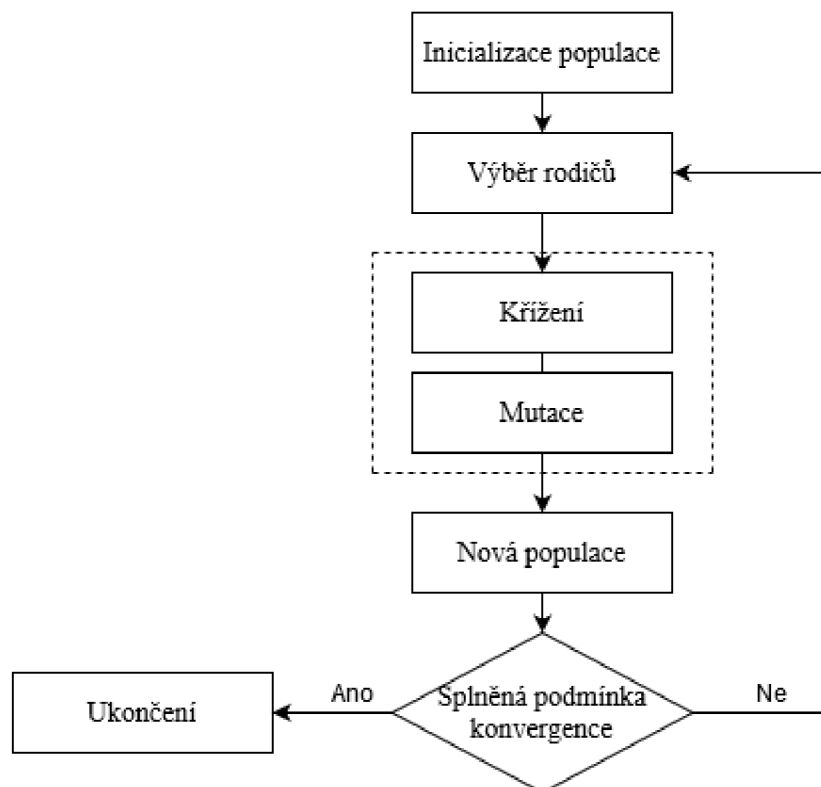
graf 1.a

Graf 1.a se skládá ze dvou funkcí. V intervalu $(0; 4)$ vidíme funkci $\sin(x)$, v intervalu $(4; 9)$ je vykreslena funkce $2\sin(x)$. Červeným a modrým křížkem jsou označeny lokální minima, globálním minimem je zřejmě červený bod $[4.71; -2]$. V případě užití klasických optimalizačních metod hrozí uvíznutí v lokálním minimu $[3.14; -1]$.

Evoluční algoritmy mají oproti jiným stochastickým metodám i klasickým optimalizačním metodám sníženou citlivost na počáteční podmínky. Tato vlastnost je dána paralelním řešením úlohy (algoritmy nepracují s jednotlivým řešením, ale s celou množinou řešení). Při hledání optimálního bodu není ani překážkou krátkodobé zhoršení řešení, což znamená, že hledaný optimální bod nemusí ležet ve směru největšího gradientu, ale dokonce i ve směru k němu opačném.

Unikátní vlastností evolučních algoritmů je možnost adaptace. Algoritmus je tedy schopen najít optimální řešení i v případě, že se během výpočtu mění.

Naopak nevýhodou evolučních algoritmů je nemožnost určit počet iterací potřebných k dosažení optima. Běžně se při spouštění, kromě požadované přesnosti, zadává maximální počet iterací, což nezaručuje nalezení řešení.



graf. 1.b

V grafu 1.b vidíme algoritmus optimalizace s užitím gramatické evoluce. V čárkovaném obdélníku jsou zaznačeny operace, které vznikají užitím genetických algoritmů.

1.1. Genetické algoritmy

Genetické algoritmy jsou skupinou algoritmů patřících mezi evoluční algoritmy a řadí se mezi algoritmy prohledávací. Hledají cílový stav splňující omezující podmínku a zároveň obvykle poskytující minimální nebo maximální hodnotu objektivní funkce. Dále nepotřebují znát přesný tvar optimalizované funkce, ale stačí znalost velikosti hodnot objektivní funkce. Tyto hodnoty porovná kriteriální funkce a určí případně lepší stav.

Mezi výhody těchto algoritmů patří snížení matematických požadavků na optimalizovanou úlohu. V některých případech, je možno, použitím genetických algoritmů, optimalizovat úlohu, která není dána funkcí (např. empiricky získané hodnoty). Genetické algoritmy můžeme použít i v případě, že neznáme počáteční podmínky, tzn. počáteční stav je zcela náhodný. V takovém případě se zvyšuje výpočetní čas. Další významnou výhodou je možnost prohledávat stavový prostor, až se stovkami stavů, ve více směrech zároveň bez potřeby dalších pomocných údajů. Ovšem pomocné údaje mohou optimalizaci výrazně urychlit.

Naproti tomu, řešení je pouze numerické, proto jej nemůžeme použít na stejnou úlohu s odlišnými parametry. Stejně jako u ostatních evolučních algoritmů, i v tomto případě, je problém s ukončovacím kritériem a s nalezením přesného řešení. Dalším problémem je časová náročnost nebo obtížný návrh kriteriální funkce.

Pro použití je nutno zvolit vhodné kódování. Toto je závislé na povaze řešené úlohy, především tedy na jednotlivých proměnných.

1.2. Pojmy

- Jedinec/hypotéza: jedno z řešení úlohy. Je složen z genotypu a fenotypu.
- Fenotyp: vlastní řešení úlohy. Jedná se o souhrn vnějších znaků a vlastností jedince.
- Genotyp: kompletní genetická výbava jedince.
- Genom: množina chromozomů jedince. V genetických algoritmech se velmi často shoduje s genotypem.
- Chromozom/kodón: vektor genů. Část genetické výbavy. Genetické algoritmy obvykle používá pouze jeden chromozom, ten potom odpovídá genomu a zároveň genotypu. Tohoto principu se budeme držet i v této práci a můžeme tvrdit, že chromozom je *kompletní genetická výbava jedince*.

$$\vec{c} = (a_1, a_2, \dots, a_k)$$

kde: k je počet genů;

a_i je daná alela.

Gen: jednotka genetické výbavy. Vychází z použitého kódování a řešeného problému.

Alela: konkrétní hodnota genu. Obvykle se jedná o přirozené číslo, které pomocí matematických operací určí pořadové číslo pravidla gramatiky.

$$a \in \mathbb{N}$$

1.2.1. Příklad určování chromozomu (genomu/genotypu) jedince

Vezměme objekt skládající se ze tří parametrů: tvar, velikost, barva. Tyto parametry jsou jednotlivé alely v daném pořadí ($a_1 = \text{tvar}$, $a_2 = \text{velikost}$, $a_3 = \text{barva}$). V tabulce (tab. 1.2.1.a) vidíme přípustné varianty jednotlivých parametrů.

<i>tvar</i>		<i>velikost</i>		<i>barva</i>	
<i>varianta</i>	<i>hodnota alely</i>	<i>varianta</i>	<i>hodnota alely</i>	<i>varianta</i>	<i>hodnota alely</i>
trojúhelník	1	malá	1	červená	1
čtyřúhelník	2	střední	2	žlutá	2
pětiúhelník	3	velká	3	zelená	3

tab. 1.2.1.a – možné parametry objektu a hodnoty alel

Genom určíme jako vektor alel:

<i>objekt</i>	<i>alely</i>	<i>genom</i>
	$a_1 = 1$	
středně veliký, žlutý, trojúhelník	$a_2 = 2$	$c_1 = (1; 2; 2)$
	$a_3 = 2$	

tab. 1.2.1.b – určování genomu objektu

Z definice vektoru je zřejmé, že nelze zapsat alely v libovolném pořadí. Vezměme: c_1 a zapišme je v opačném pořadí:

$$c_1 = (1; 2; 2) \Rightarrow c_1^* = (2; 2; 1)$$

Interpretací dle tab. 2.2.1. a získáme: červený, středně veliký čtyřúhelník.

Populace: množina všech jedinců/možných řešení, které jsou v čase t k dispozici. Nejvhodnější jedinci se vyberou a jsou dále rozvíjeni. Populaci můžeme mít současně více a dále je strukturovat. Mezi jednotlivými populacemi pak dochází k výměně informací, které se na ostatní populace mohou aplikovat. Výhodou strukturované populace je větší rozmanitost, která snižuje riziko předčasné konvergence.

$$P_t = \{\vec{h}_1, \vec{h}_2, \dots, \vec{h}_3\}$$

kde: \vec{h}_i je daný jedinec

Fitness: hodnocení každé hypotézy/každého jedince vytvořeného během výpočtů. Přiřazuje se pomocí kriteriální funkce, a to buď na konci nebo na začátku každého populačního cyklu. Může se jednat i o vzdálenost jedince od optima.

Pro vyhodnocení fitness je tedy nutné znát pouze kriteriální funkci nikoli přesný matematický model řešeného problému. Kriteriální funkce určuje lepší a horší řešení na jejímž základě se porovnávají jednotlivé hypotézy.

Operátory: postupy vyvíjející populaci, bez znalosti kriteriální funkce. Běžně se jedná o selekci (výběr), mutaci (změnu genetické informace jedince), kopírování a křížení. Příklady a vysvětlení operátorů nejsou detailněji uvedeny z důvodu širším zabývání v kapitole 3. Popis algoritmu PonyGE2.

Populační cyklus: optimalizační děj, který se opakuje tak dlouho, dokud není dosaženo požadovaného fitness. Během cyklu dojde k vytvoření nové populace, využitím operátorů. Na konci každého cyklu, resp. na začátku, se určuje hodnota fitness. (Jak již bylo zmiňováno výše.)

Typy cyklů: paralelní;
cyklické;
stromové;
hierarchické;
úplné.

Stagnace populace: stav po dokončených m cyklech. Populace se již dále nevyvíjí, to znamená, že maximální fitness se nezvyšuje. Můžeme interpretovat jako uvíznutí v lokálním minimu. Stagnace populace bývá také využíváno jako jedno z ukončovacích kritérií.

1.1. Variační operátory

Tyto operátory se GA používají pro prozkoumávání prostoru změnou genetických údajů jedinců. Díky tomu mohou prohledávat nové oblasti.

2. Gramatická evoluce

Gramatická evoluce je spojením gramatiky a genetických algoritmů s velice širokým využitím. Unikátní vlastností je vyjádření symbolického modelu řešení úlohy zvolenou gramatikou. Gramatická evoluce poskytuje, kromě výsledného řešení, také postup, kterým je řešení určeno.

Výhodou je popsateľnost složitých struktur jako jsou grafy, sítě, chemické struktury atd. V tomto přístupu se genetické algoritmy používají při hledání vhodného odvození řešení formální gramatikou.

2.1. Pojmy

Slovo: necht' *abeceda* V je neprázdná množina znaků (symbolů). Slovo w nad *abecedou* V je konečná posloupnost tvořena znaky z V .

$$w = \{a_i\}_{i=1}^n; a_i \in V$$

kde n je délka slova $|w| = n$. Prázdné slovo ϵ má délku $n = 0$.

Jazyk: jazyk L nad *abecedou* V je libovolná množina slov z V . V^* označuje množinu všech slov včetně prázdného slova $\epsilon \in V^*$.

Přepisovací pravidlo: uspořádaná dvojice (u, v) , kde $u, v \in V^*$. Slovo u se přepíše na slovo v a značí se $u \rightarrow v$.

Slovo u se přepíše přímo na z ($u \Rightarrow_R z$), pokud existují slova $v, w, x, y \in V^*$ taková, že platí $u = vxw$, $z = vyw$ a $x \rightarrow y$ je přepisovací pravidlo $p_1 \in P$.

Slovo u se přepíše na z ($u \Rightarrow_R^* z$), pokud existuje posloupnost slov $u_1, u_2, \dots, u_n; n \geq 1; u \in V^*$ taková, že platí:

$$u = u_1 \Rightarrow_R u_2 \Rightarrow_R \dots \Rightarrow_R u_n = z.$$

Posloupnost se nazývá odvozením slova z ze slova u a jeho délka je n .

Přepisovací systém: uspořádaná dvojice $R = (V, P)$, kde V je konečná abeceda, P je konečná množina přepisovacích pravidel.

Gramatika: soustava pravidel generující jazyk.

$$G = (N, \Sigma, P, S),$$

N je množina (abeceda) neterminálních symbolů (proměnných);

Σ je množina (abeceda) terminálních symbolů (konečných symbolů);

P je množina přepisovacích pravidel ve tvaru $\alpha \rightarrow \beta$, $\alpha, \beta \in (N \cup \Sigma)^*$;

S je počáteční znak neterminál.

Kontextová gram.: formální gramatika, kde pravidla P jsou tvaru:

$$\alpha A \beta \rightarrow \alpha \gamma \beta$$

kde: A – je neterminál;

β, α – jsou řetězce terminálů a neterminálů;

γ – je neprázdný řetězec terminálů a neterminálů.

2.2. Backus-Naurova forma

Backus-Naurova forma je systém zápisu bezkontextových gramatik původně vytvořenými pro jazyk Algol. Běžně se využívá pro zápis sad instrukcí, gramatik programovacích jazyků atp. Je výhodná díky užívání syntaktických zkratk.

Pro pravidla se používá tvar:

$$\langle \alpha \rangle ::= \beta_1 \mid \beta_2 \mid \dots \mid \beta_n$$

- kde: α – je neterminál;
 β_i – určuje varianty, na které lze neterminál α přepsat. Každá varianta je řetězcem terminálů a neterminálů;
 $::=$ – „go to“, je symbol přiřazující terminálu řetězec skládající se z terminálů a neterminálů. Každý neterminál lze v jednom, nebo více krocích substituovat terminály;
 \mid – „nebo“, je symbol oddělující možné řetězce;
 $\langle \rangle$ – symbol označující neterminál.

2.2.1. Příklad pravidel pro generování obrazců

Ukázku demonstrujeme na modifikaci příkladu 2.2.1. Jedná se o stejný princip úlohy, viz výše, zapsáno použitím pravidel Backus-Naurovy formy.

- $\langle \text{obrazec} \rangle ::= \langle \text{tvar} \rangle \langle \text{velikost} \rangle \langle \text{barva} \rangle$
 $\langle \text{tvar} \rangle ::= \text{trojúhelník} \mid \text{čtyřúhelník} \mid \text{pětiúhelník}$
 $\langle \text{velikost} \rangle ::= \text{malá} \mid \text{střední} \mid \text{velká}$
 $\langle \text{barva} \rangle ::= \text{červená} \mid \text{žlutá} \mid \text{zelená}$

2.2.2. Příklad pravidel pro generování obrazců se vznikajícími řetězci

Ukázka vznikajících řetězců terminálů a neterminálů.

- $\langle \text{obrazec} \rangle ::= \langle \text{velikost} \rangle \text{červená} \langle \text{tvar} \rangle \mid \langle \text{velikost} \rangle \text{žlutá} \langle \text{tvar} \rangle \mid \langle \text{velikost} \rangle \text{zelená} \langle \text{tvar} \rangle$
 $\langle \text{tvar} \rangle ::= \text{trojúhelník} \mid \text{čtyřúhelník} \mid \text{pětiúhelník}$
 $\langle \text{barva + vel.} \rangle ::= \langle \text{velikost} \rangle \text{červená} \mid \langle \text{velikost} \rangle \text{žlutá} \mid \langle \text{velikost} \rangle \text{zelená}$
 $\langle \text{velikost} \rangle ::= \text{malá} \mid \text{střední} \mid \text{velká}$

3. Popis algoritmu PonyGE2

PonyGE2 je freeware dostupný na GitHub. Jedná se o implementaci gramatické evoluce v jazyce python původně vytvořeného jako výchozí program pro studenty a výzkumníky k rychlému testování vlastních experimentů. Vychází ze souboru PonyGE, který se rozrostl do dnešní podoby.

3.1. Gramatika

V algoritmu ponyGE2 je užitá Backus-Naurova forma gramatiky, která je popsána v kapitole 2.2. Spolu s fitness funkcí je gramatika nejnáročnější na specifikaci v každém optimalizačním softwaru. Volba gramatiky velmi ovlivňuje výkon algoritmu PonyGE2.

3.2. Chromozomová reprezentace, derivační strom a inicializace

GE používá lineární chromozomy pro kódování genetické informace, ty jsou popsány zvolenou gramatikou, v našem případě BNF formou a produkují fenotyp. Každý jedinec má přiřazen chromozom, který může být použit pro reprodukování tohoto jedince.

Genotyp je využíván pro přepis počátečního symbolu, definovaného gramatikou, na terminály kodónu. Vygeneruje příslušné celočíselné hodnoty určující produkci pomocí modulárního pravidla. Modulární pravidlo je tvaru:

$$\text{Pravidlo} = c \% r$$

kde: c – je celočíselná hodnota kodónu;
 r – je počet variant pravidla pro daný neterminál;
 $\%$ – modulo.

Např. pravidlo přiřazující neterminálu matematický operátor přiřazuje každé variantě hodnotu od nuly do čtyř:

$$\begin{aligned} \langle \text{op} \rangle &::= + & (0) \\ &| - & (1) \\ &| * & (2) \\ &| / & (3). \end{aligned}$$

Pro inicializaci populace jedinců lze použít generování náhodných řetězců genomu. Ovšem při špatné volbě gramatiky mohou vzniknout komplikace snižující kvalitu populace. Proto je vhodnější zabývat se spíše inicializacemi derivačních stromů popsanými níže.

Derivační strom: vzniká při přepisu genotypu na fenotyp každého jedince. Skládá se z jednotlivých uzlů a větví.

Uzel: je každý krok, kdy se vybírá z jednotlivých možných variant při produkci.

Hloubka stromu: počet rodičů ve stromu. (Počet kroků, které jsou nutné k tomu, aby se každý neterminál popsal terminály.) Počáteční symbol gramatiky má definovanou hloubku 1.

Pro zkrácení doby nemusí v určitých konfiguracích vzniknout úplný derivační strom, stačí znát číselné informace o vlastnostech stromu (počet uzlů a hloubku). Jedinci potom mají stejné vlastnosti, jako ti, kteří jsou generováni z derivačního stromu. Můžeme tvrdit, že řešení je dáno derivačním stromem. Tato řešení jsou vždy platná.

Tvorbu derivačního stromu lze ovlivnit např. nastavením požadavku minimální hloubky. Strom se do určité hloubky bude tvořit pouze rekurzivně a tím se zajistí dané minimum větví. Další výhodnou vlastností je provádění operací užitím derivačního stromu. Vybereme-li ze stromu specifický podstrom (to je taková část stromu, která splňuje nějakou námi zvolenou vhodnou vlastnost: velikost, obsahuje specifický uzel atd.), můžeme ovlivnit variantu řízení operací jako jsou mutace nebo křížení.

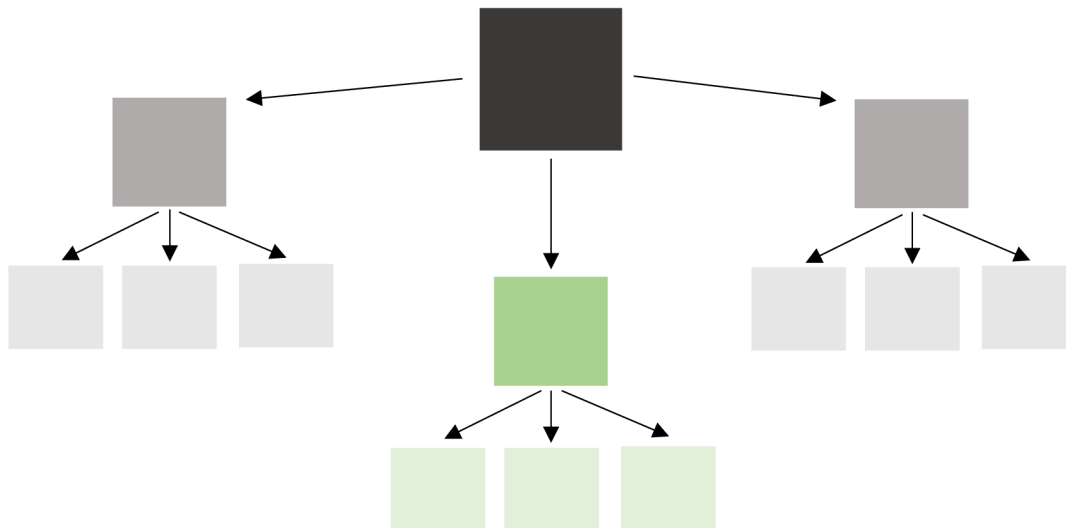


diagram 3.2.a

V diagramu 3.2.a je schéma derivačního stromu. Pokud zjistíme, že chybná řešení vznikají ve větví vyznačené zeleně, můžeme vybrat pouze tento specifický podstrom a pomocí operátorů upravovat řešení.

V algoritmu PonyGE2 jsou implementovány tři způsoby inicializací derivačního stromu: náhodná inicializace stromu, Ramped Half-and-Half/Sensible Initialisation a inicializace růstu nezávislá na pozici.

3.2.1. Náhodná inicializace stromu

Náhodná inicializace derivačního stromu generuje jednotlivce náhodně, maximálně do požadované hloubky stromu. V algoritmu neexistují žádné požadavky, které by způsobily generování maximálních hloubek derivačních stromů. Tzn., že stromy budou mít náhodnou velikost. Vzhledem k povaze gramatiky se generují spíše menší derivační stromy.

3.2.2. Ramped Half-and-Half/Sensible Inicializace

Tato metoda inicializuje jedince dvěma různými způsoby (metodou růstu a úplnou metodou). Při použití úplné metody každá větev derivačního stromu přesahuje zadaný limit hloubky stromu a vznikají velmi husté ale rovnoměrné stromy. Naproti tomu, metoda růstu generuje náhodně vytvořený derivační strom, v němž žádná větev nepřesáhne požadovanou hloubku. Tato metoda je analogická s náhodnou inicializací stromu (viz 3.3.1).

Užitím metody RHH může v počátcích populace vznikat mnoho duplicitních jedinců. Minimalizace toho problému lze dosáhnout opatrnou volbou gramatiky.

3.2.3. Inicializace růstu nezávislá na pozici

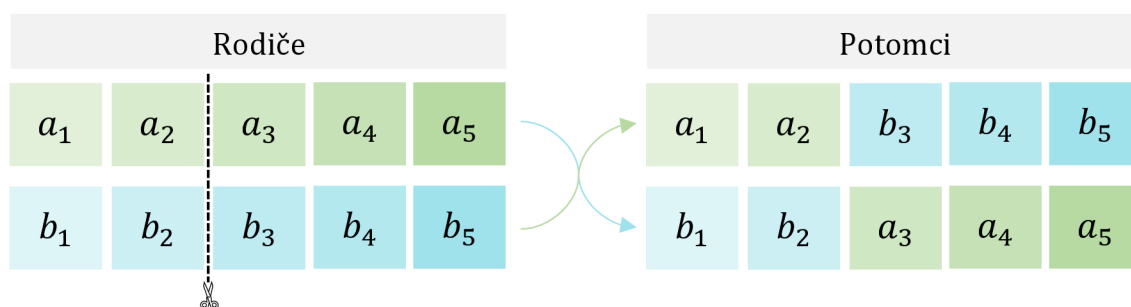
Tato metoda je obrazem metody „Ramped Half-and-Half/Sensible inicializace“. Pro generování jedince využívá pouze metodu růstu. Zároveň alespoň jedna větev stromu dosáhne požadované hloubky a neterminály jsou rozmístěny náhodně.

3.3. Křížení

Křížení je změna genetické informace záměnou části rodičovských genů. V PonyGE2 se opakuje, dokud potomek nedosahuje požadované velikosti. Jsou implementovány čtyři lineární operátory křížení a jeden operátor křížového odvození, který je odvozený z derivačního stromu. U křížového operátoru nevzniká generování neplatných jedinců, jak je tomu u lineárních operátorů.

3.3.1. Pevné jednobodové křížení

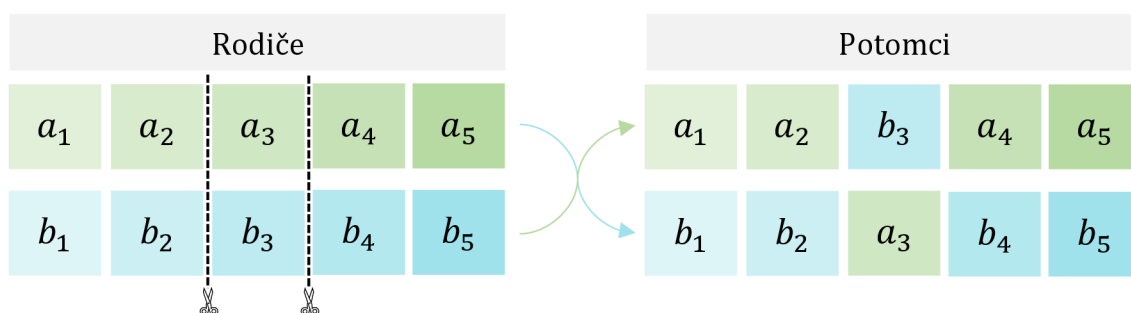
Tento operátor vezme dva rodičovské body a vytvoří z nich nového jedince způsobem zalomení a prohození hlav a ocasů jedinců. Potomci tak mají stejnou velikost jako rodičovské body.



obr. 3.3.1.a – schéma změny genetické výbavy jedince metodou pevného jednobodového křížení

3.3.2. Pevné dvoubodové křížení

Oproti předchozímu operátoru se liší pouze v zalomení rodičovských bodů ve dvou místech. Potomci opět mají stejnou velikost jako rodiče.



obr. 3.3.2.a – schéma změny genetické výbavy jedince metodou pevného dvoubodového křížení

3.3.3. Variabilní jednobodové/dvoubodové křížení.

Operátor opět ze dvou rodičovských bodů vytvoří dva potomky, ovšem nedochází k zalomení ve stejné části rodičovských chromozomů. Proto jsou potomci jiné velikosti. U jednobodového křížení dochází k jednomu zalomení, u dvou bodového kříže dochází ke dvěma zalomením rodičovského chromozomu.

3.4. Mutace

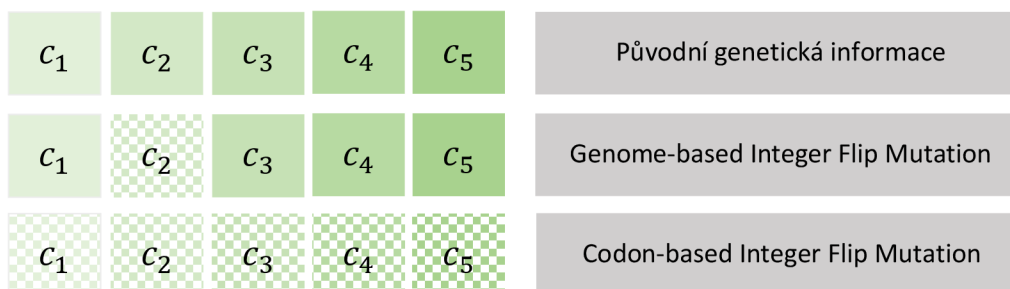
Mutace je dalším operátorem měnící genetickou informaci. Na rozdíl od selekce působí na každého jedince v populaci a nezáleží tedy na volbě rodičovských bodů. Uvedené příklady mutace jsou opět implementovány v algoritmu PonyGE2.

3.4.1. Codon-based integer Flip mutace

Tento operátor náhodně mutuje každý chromozom v genomu s danou pravděpodobností.

3.4.2. Genome-based integer Flip mutace

Mutuje chromozom náhodně vybraný z genomu



obr. 3.4.a – rozdíl mezi genomovou a kodónovou mutací. Mutované chromozomy jsou vyznačeny vzorkem.

3.5. Selektce

Algoritmus PonyGE2 má implementované dva operátory selekce/výběru.

3.5.1. Turnajová selekce

Tento operátor náhodně vybere jedince z celé populace a vrátí ty nejlepší, tzn., že s předdefinovanou pravděpodobností p se vrátí jedinec s vyšší hodnotou fitness, než byl původní vybraný. V tomto bodě nemusí nejlepší jedinec vést k optimálnímu řešení. (Analogií je jedinec pod převisem a jiný jedinec momentálně s nižším hodnocením, ale na správné cestě). Probíhá, dokud je možno vybírat nejlepší jedince.

Turnajová selekce může, ale nemusí používat nadřazování. Nepoužijeme-li nadřazování, velikost populace je rovná plné velikosti. Pokud nadřazování použijeme velikost populace je rovna plné velikosti mínus počtu elit. Toto zabrání generování dalších jedinců a následnému prohledávání.

3.5.2. Výběr zkrácením

Vytřídí celou populaci a vrátí daný nejvyšší podíl, který je nastaven dle hodnoty fitness.

3.6. Záměna populace

Operátor záměny definuje, kteří rodiče a potomci postoupí do další populace. Způsob záměny je volen dvěma následujícími způsoby:

3.7.1. Generační záměna s nadřazováním

Nahrazuje celou populaci nově generovanou. Nejlepší jedinci v populaci jsou kopírováni beze změny. Nadřazování umožňuje v každé generaci aktualizovat nejlepší řešení a tím zajišťuje nejlepší možné řešení v každém bodě algoritmu.

3.7.2. Steady State záměna

V tomto modelu nahrazují noví jedinci jedince staré bez ohledu na to, zda jsou lepší než jedinci původní. Díky generování vždy dvou nových jedinců (viz. křížení) dochází ke dvojí záměně, tedy strategii odstranění typu 2.

3.8. Vyhodnocení populace

Fitness funkce jsou ukládány v souboru fitness a jsou vždy navrhovány pro konkrétní případ. Hodnota této funkce pro optimálního jedince konverguje k nule, tedy nejlepší je ten s nejnižší hodnotou fitness. V následující kapitole naleznete konkrétní popis výpočtu fitness funkce.

4. Testovací úloha algoritmu PonyGE2

Do testovací úlohy byl využit ukázkový příklad pro generování tvarů, velikosti a barvy rozšířený o teplotu. Teplota souvisí s barvou obdobně, jako v termo kameře. Po pěti stupních se mění barva od modré, přes zelenou, žlutou a oranžovou až po barvu červenou. Navíc gramatika obsahuje další dva tvary (šestiúhelník a kruh). Jak již bylo zmíněno výše, algoritmus využívá Backus-Naurovy formy. Pro větší složitost testovací úlohy, je pro generování hodnoty teploty využíváno zobrazení $\varphi: \mathbb{N}^n \rightarrow \mathbb{R}$ definováno jako: libovolný počet $m \in \mathbb{N}$ se binárními operacemi (sčítáním, odčítáním, násobením a dělením) zobrazí do \mathbb{R} tak, že operace sčítání a odčítání mají vyšší prioritu (sčítání resp. odčítání mají přednost před násobením resp. dělením). Konkrétní užitá gramatika vypadá následovně:

```
1 <expr>::=<velikost>,<barva>,<tvar>,<teplota>
2
3 <tvar>::=trojuhelnik|ctyruhelnik|petiuhelnik|sestiuhelnik|kruh
4 <velikost>::=maly|stredni|velky
5 <barva>::=modry|zeleny|zluty|oranzy|cerveny
6
7 <teplota>::=<cislo><operator2><teplota>|(<cislo><scitanec>|<cislo>
8 <scitanec>::=<operator1><cislo>)<operator2><teplota>|
9           <operator1><cislo><scitanec>|<operator1><cislo>)
10
11 <cislo>::=<cislice><cislo>|<cislice0>
12
13 <cislice>::=1|2|3|4|5|6|7|8|9
14 <cislice0>::=0|1|2|3|4|5|6|7|8|9
15 <operator1>::=+|-
16 <operator2>::=*|/
```

tab. 4.a – implementovaná gramatika v algoritmu. Název souboru: test_teplota.bnf

Parametry genetické evoluce jsou nastaveny v souboru test_teplota.txt uloženém v adresáři parameters. Pro vzorový příklad se nejprve počítalo s těmito parametry, následně byla tabulka modifikována viz níže.

```
CACHE:                True
CODON_SIZE:           10000                                # maximalni delka kodonu
CROSSOVER:            fixed_onepoint                       # typ krizeni
CROSSOVER_PROBABILITY: 0.75                                # pravdepodobnost krizeni
DEBUG:                False
GENERATIONS:          15                                    # pocet generaci
MAX_GENOME_LENGTH:    20                                    # maximalni delka genomu
GRAMMAR_FILE:         test_teplota.bnf                     # soubor gramatiky
INITIALISATION:       PI_grow                               # soubor inicializace
INVALID_SELECTION:    True                                  # povoleni neplatnych jedincu
MAX_INIT_TREE_DEPTH:  10                                    # max. inicial. hloubka stromu
MAX_TREE_DEPTH:       17                                    # maximalni hloubka stromu
MUTATION:             int_flip_per_ind                     # operator mutace
POPULATION_SIZE:       500                                  # velikost populace
FITNESS_FUNCTION:     _testFit                              # soubor fitness fce
REPLACEMENT:          generational                         # operátor záměny
SELECTION:            tournament                           # typ vyuzite selekce
TARGET:               13                                    # soucet hodnot tvaru
TOURNAMENT_SIZE:      2                                    # pocet jedincu
VERBOSE:              False
```

tab. 4.b – soubor nastavených parametrů. test_teplota.txt

Výpočet fitness pro užitou gramatiku je uložena v souboru `_testFit.py` vypadá následovně: gramatika obsahuje pět tvarů, které jsou hodnoceny kvalitou od $\xi \in \{1, 2, 3, 4, 5\}$, čím více úhlů tvar má tím větší hodnotu kvality ξ nabývá. Obdobně jsou hodnoceny velikosti, barvy a teplota s tím rozdílem, že barva navíc závisí na teplotě, tedy hodnota barvy a teploty musí nabývat stejného ξ_i . Obrazec může nabývat tři velikostí: malý, střední a velký (s rostoucí velikostí, roste hodnota ξ). K teplotě v intervalu (0,5) odpovídá modrá barva a $\xi_i = 1$, interval (5,10) má být spojen se zelenou barvou a hodnotou $\xi_i = 2$, dále pro teplotu z (10,15) je volena barva žlutá a $\xi_i = 3$, k oranžové barvě jsou přidruženy hodnoty teploty interval (15,20) a odpovídající $\xi_i = 4$, a nakonec pro červenou barvu jsou platné teploty (20,25) a $\xi_i = 5$. Snažíme se dosáhnout generovaného obrazce: veliký, červený, kruh s teplotou v intervalu (20,25). Jehož hodnota je součtem tvaru a velikosti: *velký* + *kruh* + *červený* = 3 + 5 + 5 = 13 (proto je v parametrech nastavena hodnota cíle: *TARGET*: 13). Fitness funkce vyhodnocuje, jak daleko je generovaný obrazec od cíle, penalizuje nesoulad mezi barvou a teplotou (při splnění podmínky je rovno 0), vzdálenost od nejvyšší teploty (ve ideálním případě opět rovno 0) a dobu trvání hledání objektu. Postupně by fitness funkce měla konvergovat k nule. Výstupem z `_testFit.py` je samotná hodnota fitness každého jedince (tedy vzdálenost od nuly). Následující zdrojový kód obsahuje výpočet funkce fitness:

```

1 from algorithm.parameters import params
2 from stats.stats import stats
3 from fitness.base_ff_classes.base_ff import base_ff
4
5
6 class _testFit(base_ff):
7
8     def __init__(self):
9
10        super().__init__()
11
12        self.target = params['TARGET']           # načtená hodnota požadovaného
13                                                # cíle, požadovaná kvalita
14    def evaluate(self, ind, **kwargs):
15        guess = ind.phenotype                   # načtení fenotypu
16        vypocet = guess.split(",")              # převedení fenotypu do tabulky
17                                                # oddělí velikost, barvu, tvar a teplotu
18
19        vypocet[3]=eval(vypocet[3])              # převedení řetězce teploty
20                                                # na číselnou hodnotu
21
22        if vypocet[0] == "malý":                 # přiřazování hodnot kvality
23            vypocet[0]=1                         # objektu
24        elif vypocet[0] == "stredni":
25            vypocet[0]=2;
26        else:
27            vypocet[0]=3
28
29        if vypocet[1] == "modry":
30            vypocet[1]=1
31        elif vypocet[1] == "zeleny":
32            vypocet[1]=2;
33        elif vypocet[1] == "zluty":
34            vypocet[1]=3;
35        elif vypocet[1] == "oranzovy":
36            vypocet[1]=4;
37        else:
38            vypocet[1]=5;

```

```

39
40     if vypocet[2] == "trojuhelnik":
41         vypocet[2]=1
42     elif vypocet[2] == "ctyruhelnik":
43         vypocet[2]=2;
44     elif vypocet[2] == "petiuhelnik":
45         vypocet[2]=3;
46     elif vypocet[2] == "sestiuhelnik":
47         vypocet[2]=4;
48     else:
49         vypocet[2]=5
50
51     vypocet[3]=int(vypocet[3])
52     if vypocet[3]<=5:
53         vypocet[3]=1
54     elif vypocet[3]>5 and vypocet[3]<=10:
55         vypocet[3]=2
56     elif vypocet[3]>10 and vypocet[3]<=15:
57         vypocet[3]=3
58     elif vypocet[3]>15 and vypocet[3]<=20:
59         vypocet[3]=4
60     elif vypocet[3]>20 and vypocet[3]<=25:
61         vypocet[3]=5
62     else:
63         vypocet[3]=-5
64
65     hodnota=(vypocet[0]+vypocet[1]+vypocet[2]) # vypočet kvality objektu
66                                                # (velikosti, tvaru a barvy)
67
68     if vypocet[1]==vypocet[3]: # penalizace nesouvisející barvy s teplotou
69         penalizace_tepl=0
70     else:
71         penalizace_tepl=1.5
72
73     casova_penalizace = stats['total_time'] # penalizace za dobu hledání
74
75     fitness = (self.target - hodnota) + penalizace_tepl +
76               + casova_penalizace/1000 # výpočet fitness + penalizace
77
78     return fitness # výstup hodnoty fitness

```

tab. 4.c – výpočet hodnoty fitness pro danou gramatiku. Název souboru: *_testFit.py*

Výstupy z algoritmu se ukládají do složky results a obsahují: graf nejlepší fitness skrze jednotlivé populace, počet generací, fenotyp, genotyp, procentuální zastoupení neplatných jedinců, celkový výpočetní čas, počet neplatných jedinců a fitness funkci nejlepšího jedince.

Všechny tyto soubory jsou uvedeny v elektronických přílohách, včetně volání softwaru.

4.1. Statistika výsledků

Pro optimalizaci výše zadaného problému bylo postupně použito šest modifikací parametrů, a to záměnou operátorů. Při každé změně operátorů proběhlo padesát výpočtů, které jsou dále mezi sebou porovnávány. Operátory byly měněny dle následující tabulky (vždy pouze jeden vzhledem k tab. 4.b):

Sada1	Sada 2	Sada 3	Sada 4	Sada 5	Sada 6
Dle tab. 4.b	Křížení: fixed_twopoint	Mutace: int_flip_per_codon	Záměna: steady_state	Selekce: truncation	Křížení: Variable_onepoint

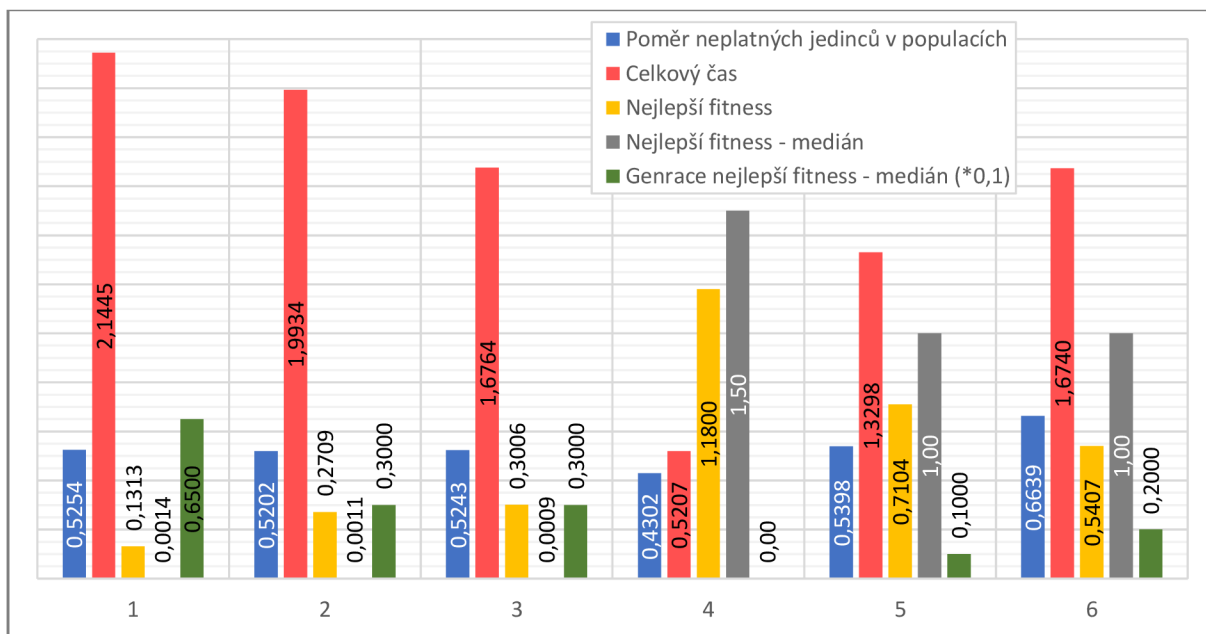
tab. 4.1.a – modifikace parametrů

Následující tabulka obsahuje informace o optimalizovaných datech. Ve čtvrté sadě se nám povedlo snížit výpočetní čas a poměrné zastoupení neplatných jedinců skrze všechny generace. Nicméně jedinci s nejlepší fitness se vyskytli v primární generaci a dále se jej nepovedlo optimalizovat, proto je průměrná hodnota fitness velmi vysoká. Dále je poměrně vysoká průměrná fitness v páté sadě a z tabulek v přílohách lze vyčíst, že bylo generováno velké množství jedinců s velmi špatnou fitness a také během nulté generace. Zaměříme-li se na medián průměrné fitness funkce, můžeme za nevyhovující považovat i šestou sadu měření.

Naopak nejlepší průměrné fitness bylo dosaženo při parametrech první sady. Nejlepší jedinci mají sice vyšší medián než při testovací sadě tři, nicméně to je způsobeno časovou penalizací. Nastavení parametrů pro nejlepší fitness je vykoupeno nejdelším výpočetním časem. Pro přehlednost je uveden graf 4.1.c, který znázorňuje průměrné hodnoty jednotlivých sad.

		1. SADA	2. SADA	3. SADA	4. SADA	5. SADA	6. SADA
Procento neplatných jedinců	průměr	52.5403	52.0260	52.4330	43.0245	53.9805	66.3903
	medián	53.0938	52.1750	52.4063	42.9625	54.1813	66.1438
Celkový výpočetní čas	průměr	2.1445	1.9934	1.6764	0.5207	1.3298	1.6740
	rozptyl	0.2189	0.0065	0.0136	0.0021	0.0053	0.0713
	medián	1.9984	1.9860	1.6864	0.5188	1.3199	1.6661
Nejlepší fitness	průměr	0.13126	0.27086	0.30063	1.18000	0.71040	0.54066
	rozptyl	0.12795	0.24188	0.23982	0.26760	0.34075	0.27829
	medián	0.00137	0.00111	0.00094	1.5	1.0	1.0
Generace s nejlepší fitness	průměr	6.34	4.26	3.62	0.0	2.94	3.04
	medián	6.5	3.0	3.0	0.0	1.0	2.0
Počet neplatných jedinců	průměr	4 203.22	4 161.28	4 194.64	3 441.96	4 318.44	5 311.22
	medián	4 247.50	4 174.00	4 192.50	3 437.00	4 334.50	5 291.50

tab. 4.1.b – zpracované hodnoty sad optimalizací

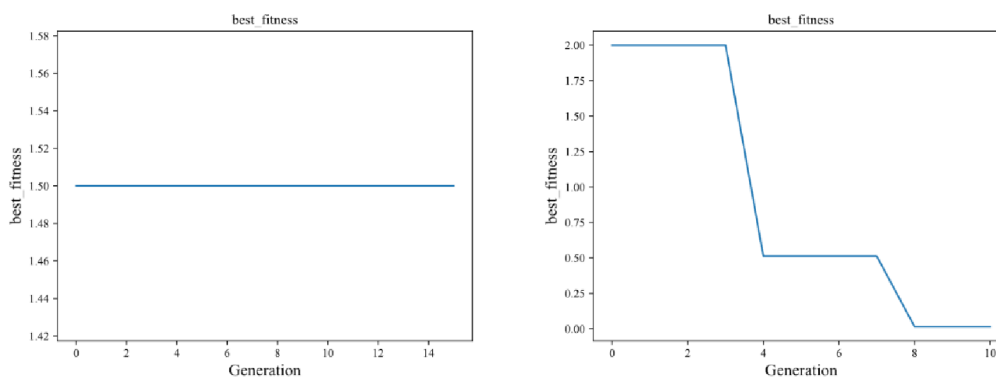


graf 4.1.c – není-li uvedeno jinak, graf znázorňuje průměrné hodnoty jednotlivých sad.

Vznik neplatných jedinců je způsoben aplikovaných operátorů. Gramatika má povoleno generovat teplotu, která je ve formátu výše popsaného výrazu. Vezmeme-li operátor křížení, který aplikujeme na daného jedince, může dojít k zalomení v „nevhodném“ bodě výrazu, tedy výraz by nesplňoval danou gramatiku. Počet neplatných jedinců je přepočítán na poměrné zastoupení skrze všechny populace (tedy na všechny vygenerované jedince). Můžeme si všimnout, že procentuální zastoupení je ve všech testovacích sadách poměrně vysoké, činí nejméně 50 % jedinců. U šesté testovací sady dokonce číslo přesáhlo 60 %. Jediná sada, kde procentuální zastoupení bylo nižší, než polovina jedinců je čtvrtá sada, která ovšem nevyhovuje, z již výše zmíněných důvodů. (V tab. 4.1.b jsou vypsány hodnoty pro procentuální zastoupení neplatných jedinců i samotný počet neplatných jedinců. Toto uvádění dvou pohledů na jednu informaci je záměrné, pro lepší představu, o jak vysokých hodnotách mluvíme bez nutnosti listovat přílohami.)

Vhodná optimalizační metoda, by měla dosahovat nejnižší hodnoty fitness a co možná nejnižšího výpočetního času. Dále by bylo vhodné snížit nebo ideálně zabránit ve tvorbě neplatných jedinců.

Pro úplnost je uveden graf 4.1.d, který je generován algoritmem PonyGE2 pro každý jeden výpočet. Graf znázorňuje nejnižší hodnotu fitness v každé generaci. Není-li v mladší generaci jedinec s nižší hodnotou fitness, je jako optimální jedinec brán ten, který v nulté generaci dosahuje minima a grafem je přímka. Pokud se v mladších generacích vyskytne jedinec s nižší hodnotou fitness, grafem je lomená čára (ideálně konverguje k nule).



Graf 4.1.d – grafy nejlepší fitness vygenerované algoritmem PonyGE2.

5. Jádrové metody

Jedná se o skupinu algoritmů využívaných ve strojovém učení pro analýzu vzorů. Tyto metody, na rozdíl od jiných, vyžadují pouze definované jádro.

Většina stochastických vyhledávacích algoritmů pracují na implicitním předpokladu, že „podobné“ body mají tendenci mít podobnou hodnotu fitness. Funkce jádra pro konkrétní problém je vytvořena učením se na základě explicitního a konkrétního modelu podobnosti. Může být brána jako funkční podobnost.

5.1. Jádrová funkce

Jádrová funkce je zobrazení $k: S \times S \rightarrow \mathbb{R}$ a je s ní spojená transformace $\Phi: S \rightarrow V$.

kde: S – je původní (originální) prostor;

V – je charakteristický prostor jádra, (Hilbertův prostor);

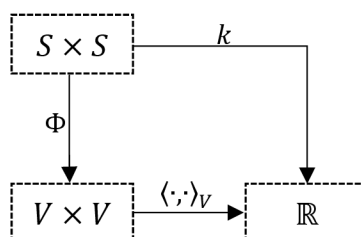
\mathbb{R} – je množina reálných čísel.

Pro transformaci platí:

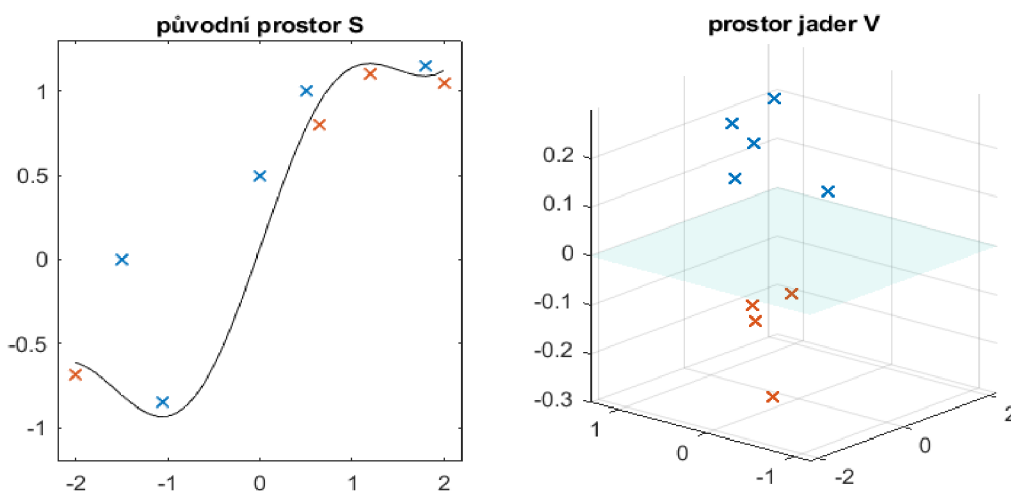
$$\Phi(s) = \{\phi_1(s), \phi_2(s), \dots\}, s \in S, \phi_i(s) \in \mathbb{R} \forall i$$

Jádrová funkce je tedy zobrazení mezi dvěma hledanými prostory, která transformuje body následovně:

$$k(s, t) = \langle \Phi(s), \Phi(t) \rangle_V = \sum_i \phi_i(s) \phi_i(t).$$



graf. 5.1.a – platí pro jádro k , Hilbertův prostor V a zobrazení Φ .



graf. 5.1.b – transformace z prostoru S do prostoru jader vyšší dimenze V .

V grafu 5.1.b je znázorněná transformace $S \rightarrow V$. Smyslem je oddělit „červené“ body od bodů „modrých“. V prostoru S jsou odděleny křivkou popsanou funkcí $y = \frac{\exp(x-2) + \sin(2x) + x}{2}$, zatím co po transformaci do prostoru V jsou body odděleny rovinou $z = 0$, kterou lze najít snadněji.

Transformace se obvykle provádí z prostoru S do prostoru V , jehož dimenze je ostře větší: $\dim(S) > \dim(V)$. Důvod transformace do prostoru vyšší dimenze je linearizace řešeného problému. Ilustraci vidíme na obrázku 5.1.b.

Funkce jádra musí splňovat následující kritéria: spojitost, být symetrická a také splňovat. Pokud jsou podmínky, splněny je zaručena transformace z S na nějaký přidružený Hilbertův prostor V , ve kterém zobrazení $k(s, t)$ udává vnitřní hodnotu.

Na funkce jádra můžeme nahlížet jako na funkci fitness. Funkce jádra je částečně konkrétní a explicitní model toho, jak moc si fitness mohou být podobné (měla by zachytit kovarianci fitness mezi body prohledávaného prostoru). Ideální jádrová funkce by se rovnala:

$$k(s, t) = \text{cov}(f(s), f(t)),$$

kde se kovariance počítá zprůměrováním přes všechny fitness funkce f .

5.2. Základní jádrové operace a mutace

Jak již bylo vysvětleno výše, jádrové funkce určují vnitřní hodnoty mezi prostory bodů a prostorem jader, do kterého jsou body transformované. Základní využívané algebraické operace pro určování hodnot jader a následné mutace a křížení jsou popsány níže.

5.2.1. Norma $\|s\|$

Podoba v prostoru jader: $\sqrt{\langle \Phi(s), \Phi(s) \rangle} = \sqrt{\sum_i \phi_i(s)^2}$.

Funkce jádra: $\sqrt{k(s, s)}$.

Fitness: $\sqrt{\text{var}(f(s))} = \sqrt{\text{cov}(f(s), f(s))}$.

5.2.2. Vnitřní hodnoty $\langle s, t \rangle$

Podoba v prostoru jader: $\langle \Phi(s), \Phi(s) \rangle = \sum_i \phi_i(s)^2$.

Funkce jádra: $k(s, s)$.

Fitness: $\text{cov}(f(s), f(s))$.

5.2.3. Vzdálenost $d(s, t)$

Podoba v prostoru jader: $\sqrt{\sum_i (\phi_i(s) - \phi_i(t))^2} =$
 $= \sqrt{\sum_i \phi_i(s)^2 + \phi_i(t)^2 - 2\phi_i(s)\phi_i(t)}$.

Funkce jádra: $\sqrt{k(s, s) + k(t, t) - 2k(s, t)}$.

Fitness: $\sqrt{(f(s) - f(t))^2}$.

5.2.4. Kvadratická vzdálenost $d^2(s, t)$

Podoba v prostoru jader: $\sum_i (\phi_i(s) - \phi_i(t))^2 =$
 $= \sum_i \phi_i(s)^2 + \phi_i(t)^2 - 2\phi_i(s)\phi_i(t)$.

Funkce jádra: $k(s, s) + k(t, t) - 2k(s, t)$.

Fitness: $(f(s) - f(t))^2$

5.2.5. Gaussova mutace

$$P_{GKM}(c) \propto \exp\left(-\frac{d^2(p, c)}{2\sigma^2}\right)$$

Kde: p – rodičovský bod;

c – nový vygenerovaný pořizovaný bod.

Gaussova mutace se používá pro většinu operátorů tohoto typu genetických algoritmů. Konstrukce operátoru Gaussovy mutace vychází z klasické eukleidovské vzdálenosti.

5.3. Lineární kombinace, hyperplochy a bázové prostory

Vytváření lineárních kombinací, hyperploch a bázových prostorů patří mezi pokročilejší operace využívané pro optimalizace.

Rodičovský prostor: $P = \{p_1, \dots, p_n\}; p_i \in S$. Množina n rodičovských bodů.

Rodičovský bázový prostor: množina S_P všech možných lineárních kombinací z nadřazených souřadnic V .

$$S_P = \text{obol}(S) = \left\{ \sum \omega_i p_i \in V \mid p_i \in P, \omega_i \in \mathbb{R} \right\}$$

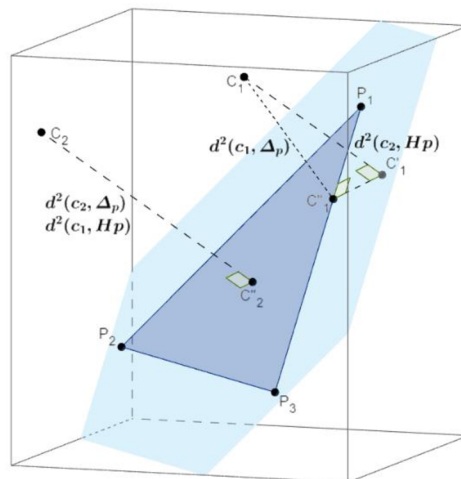
Jiná varianta: S_P je prostor protínající rodiče a počátek.

S_P^\perp je doplněk S_P do V .

Rodičovská hyperplocha: rovina H_P ve V , která protíná n rodičovských bodů.

Jiná varianta: H_P je prostor protínající pouze rodiče.

Rodičovský simplex: je plocha ohraničená rodičovskými body.



graf 5.3.a – transformace do prostoru H_P , Δ_P a zobrazení d^2 .

V grafu 5.3.a jsou zobrazeny tři rodičovské body P_1 , P_2 a P_3 . Prostor S_P je vyznačen krychlí, prostor H_P je plocha daná rodičovskými body a simplex Δ_P je trojúhelník, který vzniká spojením těchto bodů. Lze vidět, že dimenze prostoru H_P je o jedna menší, než dimenze prostoru S_P .

5.4. Souřadnicový systém

Souřadnicový systém	Základní prostor	Výpočet souřadnic
Jádrový	$\Psi_K = K_{P,P}^{-1}P$	$K_{P,c}$
Rodičovský	$\Psi_P = P$	$K_{P,P}^{-1}K_{P,c}$
Ortonormální	$\Psi_E = K_{P,P}^{-\frac{1}{2}}P$	$K_{P,P}^{-\frac{1}{2}}K_{P,c}$

tab. 4.4.a – srovnání souřadnicových systémů

Matice $K_{P,P}$: je dimenze $n \times n$ a konstruována $(K_{P,P})_{i,j} = k(p_i, p_j)$.

Vektor $K_{P,c}$: je dimenze $1 \times n$ a konstruován $(K_{P,c})_i = k(p_i, c)$, kde $c \in S$ libovolně.

5.4.1. Jádrový souřadnicový systém

Nejjednodušší souřadnicový systém, který popisuje prostor S_P . Konstruujeme užitím vektoru vnitřních hodnot s rodičovskými body P_i .

Souřadnice bodu $c \in S_P$: $[c]_K = K_{P,c}$

5.4.2. Rodičovský souřadnicový systém

Rodičovský souřadnicový systém je potenciálně užitečnější než jádrový souřadnicový systém. Jako základní vektory využívá samotné rodičovské vektory.

Souřadnice bodu $c \in S_P$: $[c]_P = K_{P,P}^{-1}K_{P,c}$

5.4.3. Ortonormální souřadnicový systém

Základní prostor: $\Psi_E = \{e^{(1)}, \dots, e^{(n)}\} = ZP$.

Kde: $e^{(i)}$ jsou ortonormální vektory;

matice $Z = QD^{-\frac{1}{2}}Q^{-1}$, D je diagonální matice vlastních hodnot a Q je sloupcový vektor vlastních vektorů matice $K_{P,P}$;

P je vektor rodičovských bodů.

Platí: $\|e^{(i)}\| = k(e^{(i)}, e^{(i)}) = 1$ pro $\forall i$;

$\langle e^{(i)}, e^{(j)} \rangle = k(e^{(i)}, e^{(j)}) = 0 \Leftrightarrow i \neq j$.

Souřadnice bodu $c \in S_P$: $[c]_P = K_{P,P}^{-\frac{1}{2}}K_{P,c}$.

(Skoro)-nulové vlastní hodnoty ukazují na (skoro)-kolineární rodičovské body. Tohoto lze využít k odstranění všech kolineárně závislých rodičů před vytvořením souřadnicového systému. Tím se zmenší prostor P a urychlí výpočty.

Výhodou ortonormálního souřadnicového systému je fakt, že body z S_P jsou vždy v souladu se základními jádrovými operacemi.

5.5. Křížení

Křížení je běžně užívanou operací genetických algoritmů. V jádrových metodách se využívají křížové mutace založené na vzdálenosti bodu c od rodičovského hyperprostoru H_P nebo rodičovského trojúhelníku (simplexu) Δ_P ve V .

Kvadratická vzdálenost bodu od hyperplochy: $d^2(c, H_P)$.

Kvadratická vzdálenost bodu od simplexu: $d^2(c, \Delta_P)$.

Operátor hledání formy hustoty pro hyperplochu:

$$P_{KH}(c) \propto \exp\left(-\frac{d^2(c, H_P)}{\sigma^2}\right)$$

kde: σ – je parametr určující penalizaci od nadřazené hyperplochy ve V .

Operátor hledání hustoty pro simplex je obdobný, pouze využívá kvadratické vzdálenosti bodu c a simplexu. Tento operátor (v obou případech) kombinuje mutaci a křížení.

Závěr

V prvních dvou částech, bakalářská práce popisuje evoluční algoritmy a gramatickou evoluci. Popis těchto metod obsahuje oblast využití, klady a zápory a ilustrační příklady, na kterých jsou nejdůležitější problémy (z hlediska další části bakalářské práce) názorně popsány. Seznámení se s těmito metodami, je důležité kvůli popisu algoritmu PonyGE2, který využívá gramatickou evoluci.

Třetí část se zabývá samotným popisem již zmíněného algoritmu. Obsahuje popis funkcí, resp. operátorů, které jsou dostupné a volně modifikovatelné (algoritmus PonyGe2 je volně dostupný optimalizační nástroj). Dále tato část obsahuje vysvětlení principů některých operátorů. Především těch, které jsou používány při optimalizaci testované úlohy. Dále uvádí princip inicializace a nastiňuje výběr optima.

Následující kapitola detailně popisuje navržený optimalizovaný příklad, jeho gramatiku, volbu parametrů i zvolenou fitness funkci. Dále obsahuje hodnoty jednotlivých testovaných sad. Testovací sadou je myšleno provádění výpočtů při různé volbě parametrů a porovnávání efektivity výpočtů. Tyto výpočty jsou pro jednodušší orientaci zaneseny do grafu a slovně okomentovány. V příloze lze najít původně naměřené hodnoty, ze kterých byla data vyčtena.

Poslední kapitola obsahuje popis jádrových metod. Jedná se o matematické metody, které se využívají pro analýzu vzorů. Celá kapitola vysvětluje podstatu jádrových metod a základní algebru, kterou využívají. Dále jsou popsány složitější operátory, jako například mutace nebo křížení. Pro vysvětlení jsou generovány schématické grafy, které vždy obsahují popis daného modelovaného příkladu pro konkrétnější představu.

Dalším pokračováním této bakalářské práce, je plánovaná implementace jádrových metod do algoritmu PonyGE2. Předpokladem je rozšíření možností pro optimalizace úloh, případně urychlení výpočtového času nebo dosažení efektivnějších výsledků (např. snížení generování neplatných jedinců).

Reference

- [1] POPELKA, Ondřej. Použití evolučních a genetických algoritmů v ekonomických aplikacích. Brno, 2008. Disertační. Mendelova lesnická a zemědělská univerzita v Brně. Vedoucí práce Doc. Ing. RNDr. Jiří Šťastný, CSc.
- [2] O'NEILL, Michael a Conor RYAN. Grammatical Evolution. In: *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*. IEEE, 2001, s. 349-358. DOI: 10.1109/4235.942529. ISSN 1089-778X.
- [3] RYAN, Conor, JJ COLLINS a Michael O'NEILL. Grammatical evolution: Evolving programs for an arbitrary language. In: *European Conference on Genetic Programming*. Paris, France: EuroGP, 1998, s. 13. DOI: <https://doi.org/10.1007/BFb0055930>. ISBN 978-3-540-64360-9.
- [4] ANEJA, Kavita, Saroj a Jyoti. A Genetic Algorithm Approach to Kernel Functions Parameters Selection for SVM. In: *International Journal of Current Engineering and Technology*. Hisar, Haryana: Dept. of Computer Science G.J.U., 2013, s. 713-716. ISSN 2277 - 4106.
- [5] GAGNÉ, Christian, Marc SCHOENAUER, Michèle SEBAG a Marco TOMASSINI. Genetic Programming for Kernel-based Learning with Co-evolving Subsets Selection. In: *Dans PPSN'06*. Reykjavik, 2006, s. 1008-1017. DOI: inria-00116344.
- [6] LANE, Fergal, R. Muhammad Atif AZAD a Conor RYAN. *Principled Evolutionary Algorithm Search Operator Design and the Kernel Trick*. In: USA: GECCO, 2016, s. 1-9. DOI: 10.1109/SSCI.2016.7850204.
- [7] FENTON, Michael, James MCDERMOTT, David FEGAN, Stefan FORSTENLECHNER, Michael O'NEILL a Erik HEMBERG. PonyGE2: Grammatical Evolution in Python. In: *Workshop on Evolutionary Computation Software Systems*. Berlin, Germany: GECCO, 2017, s. 1-13. DOI: 10.1145/3067695.3082469.

Přílohy

Fenotyp	Genotyp	Procenta neplatných	Výpočetní čas	Neplatní jedinci	fitness	generace
						1. Sada
velky,cervený,kruh,22	[384, 5918, 4769, 7279, 2576, 5078, 874, 5591, 1352, 6629]	50.20	2.1270	4016	0.0011	3
velky,cervený,kruh,(27-6)	[1929, 359, 2894, 399, 7732, 7668, 1531, 9655, 4187, 4967, 1455, 9919, 4926, 279, 1341, 3482, 3092, 6775, 164]	51.61	1.9565	4129	0.0016	12
velky,cervený,kruh,25	[7954, 1064, 9859, 6074, 1619, 2318, 9190, 9637, 3685, 2488]	47.39	1.9178	3791	0.0012	7
velky,cervený,kruh,23	[6622, 9881, 6199, 2384, 2867, 5932, 5491, 7551, 1293, 7910]	48.29	2.0109	3863	0.0006	1
velky,cervený,kruh,(9+12)	[4952, 5348, 9259, 5749, 9895, 2365, 3009, 494, 4840, 914, 6147, 3155, 5412, 6534, 6602, 239, 1800, 993, 7521]	53.95	2.0350	4316	0.0018	11
velky,cervený,kruh,24	[5530, 269, 9614, 7299, 9350, 6572, 5851, 8185, 794, 1596]	53.56	2.0086	4285	0.0018	11
velky,oranžový,kruh,37/2	[2332, 8849, 7153, 8189, 2838, 5178, 488, 1267, 9807, 9601, 5441, 9997, 9022, 5456, 6314, 7297]	54.85	1.9700	4388	1.0006	1
velky,cervený,sestihelník,24	[3317, 9560, 1689, 8758, 1790, 1006, 9370, 1695, 2584, 7859]	50.48	1.9556	4038	1.0013	8
velky,cervený,kruh,24	[1213, 7772, 4044, 7499, 3791, 7390, 6364, 9307, 2854, 963]	46.06	2.0249	3685	0.0012	7
velky,cervený,kruh,(9+11+4)	[6479, 7225, 8542, 3617, 4255, 3679, 4244, 8394, 5921, 4214, 2962, 2027, 1434, 1444, 3951, 3049, 701, 299, 8981, 139, 2124, 8523, 1205, 7760, 8209]	49.84	2.0302	3987	0.0000	0
velky,cervený,kruh,63/3	[688, 2825, 6154, 6904, 5082, 6348, 3443, 9821, 2453, 6941, 2429, 7789, 8173, 6961, 1778, 2075, 8351, 4719, 292]	51.80	1.9404	4144	0.0008	2
velky,cervený,kruh,21	[4376, 9863, 9674, 4084, 8213, 7942, 2719, 915, 4741, 747, 8113, 7727, 4502, 6731, 5144, 4054, 6057, 419, 4783]	53.28	1.9201	4262	0.0018	14
velky,cervený,kruh,21	[9591, 4568, 239, 2319, 2051, 8464, 9550, 885, 5211, 1660, 1639, 6372, 7193, 4495, 6781, 1548]	48.58	1.9541	3886	0.0011	1
velky,cervený,kruh,22	[4559, 6248, 3849, 644, 8654, 6622, 4348, 5161, 2282, 5633, 1396, 5427, 5729, 4861, 6850, 9196, 3032, 1242, 6708]	52.30	1.9613	4184	0.0006	15
velky,cervený,kruh,22	[3375, 7979, 5004, 9629, 473, 6746, 4519, 5181, 2972, 8972]	55.88	2.3285	4470	0.0023	7
velky,cervený,kruh,5*5	[2401, 9995, 5154, 5809, 699, 9013, 9885, 4076, 2600, 6399, 8245, 9514, 7752, 9511, 980, 5759, 5028, 6995, 1826]	53.48	1.9712	4278	0.0014	2
velky,oranžový,kruh,19	[1412, 4358, 4023, 6359, 3596, 9702, 1035, 9915, 8919, 3694]	50.89	2.0292	4071	1.0008	1
velky,cervený,kruh,25	[3530, 8345, 1414, 3404, 9515, 4110, 496, 5979, 3085, 9820, 5481, 5711, 190, 2418, 8532, 3654]	55.70	1.9219	4456	0.0006	9
velky,cervený,kruh,(23+2-3)	[4561, 3590, 9549, 5454, 3850, 8022, 1531, 9245, 4363, 7459, 3634, 5445, 8202, 1994, 6663, 5701, 913, 2316, 8751]	52.51	1.9615	4201	0.0014	11
velky,cervený,kruh,4*1*2*3	[5237, 7358, 139, 584, 9024, 6269, 3744, 670, 2091, 7257, 6841, 6078, 5112, 2121, 1022, 8068, 2603, 2283, 9973]	56.95	3.0722	4556	0.0028	1
velky,oranžový,kruh,16	[8794, 8162, 7543, 7204, 8408, 6340, 4716, 5487, 3376, 1623, 2357, 8241, 1282, 4255, 4088, 3747, 8964, 2643, 9319]	56.01	1.9189	4481	1.0006	1
velky,cervený,kruh,22	[9042, 6458, 8244, 3634, 8738, 9072, 5149, 8437, 2162, 7846]	51.96	1.6356	4157	0.0008	11

velky,cerveny,kruh,25	[7876, 8768, 1909, 6424, 4148, 5434, 9073, 9053, 8915, 5285]	58.14	1.9325	4651	0.0006	4
velky,cerveny,kruh,150/(3+3)	[8654, 4949, 3274, 1319, 5331, 1472, 4968, 8840, 8581, 6537, 670, 6023, 2959, 6943, 6333, 2234, 9556, 9593, 2603]	53.21	1.9882	4257	0.0017	8
velky,cerveny,kruh,25	[3447, 9623, 4644, 244, 1937, 1532, 352, 9455, 3215, 6747]	50.43	2.9043	4034	0.0018	8
velky,cerveny,kruh,(81-57)	[2401, 8123, 5774, 6904, 9640, 1292, 2617, 6119, 8451, 3245, 3577, 7204, 8293, 4781, 4437, 391, 2569, 8846, 7960]	53.61	2.7110	4289	0.0021	6
velky,cerveny,kruh,4	[329, 4514, 3089, 5129, 2131, 6394, 1133, 2554, 8599, 4041]	54.53	2.0174	4362	1.5000	0
velky,cerveny,kruh,67/3	[198, 428, 8969, 6924, 8946, 8778, 7736, 7907, 1927, 7399, 4001, 589, 5673, 146, 3309, 2490, 9826, 7051, 3051]	55.15	2.0469	4412	0.0014	8
velky,cerveny,kruh,81645/3381	[7106, 5574, 9527, 7457, 7964, 5839, 1874, 2694, 2808, 5636, 88, 9380, 5119, 5713, 8316, 4696, 9290, 6502, 9778, 2771, 2300, 2268, 9570, 7287, 2945, 4421, 4007, 6896, 1540, 828, 3900, 649, 7923, 1493, 4664, 1162, 4672]	57.21	1.9552	4577	0.0000	0
velky,cerveny,kruh,(8+16)	[4753, 3623, 9964, 9779, 1339, 9651, 3148, 8102, 9584, 7204, 3141, 8411, 8736, 5996, 361, 7223, 2254, 9250, 8245]	48.33	2.0681	3866	0.0011	6
velky,cerveny,kruh,22	[3764, 482, 5124, 6994, 7928, 6758, 6103, 2889, 9042, 6906, 8330, 4809, 843, 5478, 8570, 5682, 3202, 5875, 9232]	50.18	2.7204	4014	0.0017	4
velky,cerveny,kruh,21	[2084, 1853, 2184, 4244, 4169, 3498, 901, 5857, 7771, 7173, 1291, 9389, 1793, 8031, 1939, 7302]	56.24	1.9233	4499	0.0014	9
velky,cerveny,kruh,21	[2938, 6089, 184, 1689, 7475, 2202, 4132, 3749, 6281, 2133, 9147, 2476, 8965, 1783, 80, 9801, 2858, 9251, 3594]	55.85	1.9120	4468	0.0017	11
velky,cerveny,kruh,9595/422	[2994, 4524, 8974, 6323, 8967, 6920, 9866, 7477, 8240, 1677, 8110, 9489, 6768, 8801, 9192, 3360, 3510, 4558, 4400, 9915, 45, 943, 8875, 8438, 4481, 2363, 5555, 2261, 633, 8349, 1527]	59.16	2.7489	4733	0.0000	0
velky,cerveny,kruh,21	[1048, 4343, 8529, 2804, 8627, 934, 6319, 1089, 651, 1412, 8389, 9194, 7517, 8759, 9668, 286, 8194, 3111, 1380]	53.28	4.5063	4262	0.0038	8
velky,cerveny,kruh,(31-7)	[2174, 4529, 4499, 39, 9418, 8436, 3737, 9707, 6311, 9197, 2413, 3519, 7647, 616, 5275, 9803, 6376, 9866, 627]	46.35	1.9634	3708	0.0011	6
velky,cerveny,kruh,25	[5909, 6914, 6719, 7569, 2639, 7648, 9739, 5657, 1385, 3215, 704, 1052, 2685, 2092, 7188, 9314, 2320, 252, 5476]	49.41	1.9704	3953	0.0014	9
velky,cerveny,kruh,24	[7892, 9113, 9024, 374, 3770, 7566, 8299, 2625, 6354, 8999, 7389, 5966, 3228, 332, 1682, 8313]	53.51	1.9122	4281	0.0013	9
velky,cerveny,kruh,22	[4402, 9176, 7034, 9294, 4742, 9052, 3871, 4677, 692, 5908, 6025, 7970, 9522]	58.88	1.7549	4710	0.0005	1
velky,cerveny,kruh,21	[2232, 6773, 7334, 1944, 1196, 5336, 5707, 4975, 6511, 7916, 9349, 7121, 4553, 3694, 9575, 9490, 4685, 1586, 204]	52.13	1.9258	4170	0.0012	7
velky,cerveny,kruh,23	[484, 5528, 3314, 264, 9335, 2182, 8929, 3639, 9503, 8161, 6469, 1087, 1381, 1734, 9441, 2196, 233, 2472, 735]	53.40	3.3031	4272	0.0032	15
velky,cerveny,kruh,(23+2)	[6668, 3248, 6209, 3164, 8833, 4670, 9892, 1853, 9133, 9698, 1122, 4183, 3652, 5771, 5490, 5902, 206, 8187, 4432]	56.35	2.0536	4508	0.0006	1
stredni,cerveny,kruh,7*7/2	[7164, 4366, 7894, 9199, 5841, 4179, 7617, 1488, 7473, 767, 467, 1933, 6131, 2231, 5902, 3679, 7412, 5315, 7055]	52.06	2.0418	4165	1.0012	6
velky,cerveny,kruh,65/3	[3460, 4583, 4449, 5909, 3861, 9456, 8753, 8077, 8615, 6285, 1280, 6425, 5973, 5776, 7640, 2168]	52.98	2.0477	4238	0.0019	14
velky,cerveny,kruh,3*7	[789, 6467, 9114, 1689, 2487, 6009, 1583, 4706, 2993, 8449, 9497, 3466, 2265, 6261, 1588, 1616]	50.01	2.1523	4001	0.0008	2
velky,cerveny,kruh,25	[1821, 4658, 2249, 9604, 2552, 3452, 3115, 5193, 1495, 5968, 27, 273, 3592, 4845, 8683, 6713]	42.83	2.0417	3426	0.0010	4
velky,cerveny,kruh,21	[1727, 6701, 7619, 7139, 5564, 2604, 1828, 9499, 841, 4181, 7011, 1574, 6434, 5755, 7350, 2329]	53.90	1.9606	4312	0.0016	12
velky,cerveny,kruh,96/4	[7297, 7484, 7869, 6384, 4497, 9092, 3824, 9495, 2546, 943, 7715, 8491, 9864, 9629, 9236, 6857, 7629, 992, 1113]	51.20	2.0951	4096	0.0019	14
velky,cerveny,kruh,21	[7190, 689, 4184, 6544, 9380, 3546, 1360, 3095, 5211, 1918, 6605, 897, 8051, 2728, 9650, 2619]	48.09	2.0240	3847	0.0009	3
velky,cerveny,kruh,25	[2313, 1721, 5294, 6514, 8066, 9606, 145, 6585, 1785, 7814]	55.08	1.8919	4406	0.0011	6

2. Sada

velky,cerveny,kruh,24	[5683, 2471, 1584, 7329, 2204, 9354, 8227, 9893, 2174, 1294, 8405, 9186]	52.84	1.9835	4227	0.0014	7
velky,cerveny,kruh,24	[5335, 3722, 9964, 6189, 6686, 2750, 4267, 7223, 444, 2083, 517, 7474, 6541, 6357, 9113, 4357, 3814, 8147, 7760]	52.63	2.0085	4210	0.0011	3
velky,cerveny,kruh,24	[5335, 3722, 9964, 6189, 6686, 2750, 4267, 7223, 444, 2083, 517, 7474, 6541, 6357, 9113, 4357, 3814, 8147, 7760]	52.63	2.0085	4210	0.0011	5
velky,cerveny,kruh,25	[5552, 6581, 9799, 999, 9338, 8140, 5833, 3321, 8315, 672, 71, 450, 1606]	51.50	1.9885	4120	0.0009	3
velky,cerveny,kruh,7*3	[9885, 1037, 7029, 3024, 1332, 5841, 4827, 1654, 6980, 8889, 6583, 5548, 7331, 9477, 3152, 1996, 2564, 8174, 7023]	53.18	2.0162	4254	0.0009	3
velky,cerveny,kruh,24	[7840, 4994, 2549, 449, 3242, 8904, 1630, 1985, 4854, 3705]	46.06	2.0153	3685	0.0006	1
velky,cerveny,kruh,23	[5576, 3032, 4679, 5329, 4745, 1370, 7228, 3359, 2913, 4453, 5859, 5749, 7753, 4260, 2949]	51.59	1.9010	4127	0.0015	11
velky,cerveny,kruh,47/2	[1763, 2387, 6594, 264, 4797, 1990, 6285, 8221, 3277, 7263, 3764, 201, 9282]	52.21	1.9128	4177	0.0011	6
velky,cerveny,sestihelnik,24	[9278, 6935, 3319, 8773, 9353, 4228, 1990, 3029, 7954, 6557]	53.93	2.0274	4314	1.0010	4
velky,cerveny,kruh,(6+26-7)	[9018, 2135, 3694, 239, 9451, 3027, 8376, 8293, 1722, 6784, 2269, 6153, 8756, 3356, 7235, 9643, 8607, 4648, 87]	52.04	1.9512	4163	0.0012	7
velky,cerveny,sestihelnik,25	[4618, 4472, 3404, 8288, 287, 7964, 6571, 2515, 8565, 1392, 9879, 4838, 4540, 817, 5745, 1532, 5537, 1726, 3594]	48.81	2.0448	3905	1.0006	1
velky,cerveny,kruh,24	[5053, 4952, 1094, 6014, 8909, 4032, 3916, 8613, 5794, 3898, 8586, 6956]	52.34	2.1594	4187	0.0017	9
velky,cerveny,kruh,23	[4624, 8333, 7599, 3429, 6290, 8894, 3601, 6315, 4043, 6713, 1348, 5272, 5654, 1607, 7071, 186]	49.34	2.1257	3947	0.0013	7
velky,cerveny,kruh,(4+9+8)	[1262, 2378, 4984, 754, 9196, 8391, 2334, 7996, 8652, 2451, 6169, 1310, 7226, 8037, 2278, 4423]	45.80	2.3624	3664	0.0011	4
velky,cerveny,kruh,23	[4080, 6149, 8084, 6599, 1085, 9010, 4303, 821, 5363, 8478, 716, 6329, 2910, 7547, 1026, 2336]	51.86	2.0300	4149	0.0009	3
velky,cerveny,kruh,(32-8)	[1176, 5395, 6989, 4748, 604, 1830, 4745, 2465, 8202, 1469, 7117, 38, 6511, 261, 4429, 2495, 781, 6418, 4708]	53.54	2.0440	4283	0.0000	1
velky,cerveny,kruh,6*4	[5147, 8291, 2834, 6104, 5217, 4143, 6336, 4760, 578, 513, 6154, 4046, 7357, 4646, 1462, 2365, 4726, 9119, 624]	49.48	2.0682	3958	0.0007	1
velky,oranzovy,kruh,(5+8+6)	[8049, 6050, 9248, 9694, 7918, 4551, 4945, 1144, 780, 4747, 3428, 8111, 7134, 5197, 7766, 1600]	52.70	1.8346	4216	1.0010	5
velky,cerveny,kruh,23	[1495, 7376, 3784, 3024, 6548, 4704, 8452, 41, 6093, 1526, 4915, 4460, 6331]	50.74	1.9930	4059	0.0019	15
velky,cerveny,kruh,9	[6655, 2084, 7022, 831, 9149, 9095, 2349, 3583, 8539, 9632]	55.11	2.0431	4409	1.5000	1
velky,cerveny,kruh,21	[7822, 2348, 9839, 2924, 5189, 4206, 6706, 7713, 6951, 7976, 1862, 5281, 2373, 7114, 8459, 5902]	51.70	1.9988	4136	0.0013	7
velky,cerveny,kruh,22*1	[618, 6182, 9389, 1744, 1485, 5944, 9397, 5329, 92, 2552, 2087, 521, 4791, 5098, 284, 9709, 9946, 454, 7842]	52.39	1.9523	4191	0.0006	1
velky,cerveny,kruh,(124-98-4)	[5656, 3583, 4409, 7674, 7580, 2032, 8644, 9783, 5590, 5684, 4077, 2584, 4109, 2193, 2895, 6204, 1101, 9280, 4805, 9551, 9308, 2739, 3510, 9987, 3179, 1240, 2316, 927, 8987, 7481, 13]	52.14	1.9510	4171	0.0000	0
velky,cerveny,kruh,21	[1558, 701, 7599, 3454, 5396, 9424, 8884, 9211, 1961, 5353, 6601, 6676, 5204, 8881, 1705, 1475]	56.81	1.9729	4545	0.0016	12
velky,cerveny,kruh,3*7	[1617, 3137, 5744, 1289, 1809, 3455, 9643, 6276, 5696, 9209, 8707, 1484, 1861]	45.65	1.9821	3652	0.0009	3
velky,cerveny,kruh,6	[9085, 6260, 7984, 5579, 7904, 8649, 276, 6057, 7600, 4714]	53.75	2.0102	4300	1.5000	1
velky,cerveny,kruh,24	[3901, 5453, 4664, 9179, 6989, 7976, 2278, 7225, 1464, 2330, 3714, 3711, 4680, 8559, 5554, 1669, 8802, 5120, 8772]	45.19	1.9952	3615	0.0009	3

velky,cerveny,kruh,(9955+664)/55/8	[4766, 1756, 7590, 7444, 196, 9133, 8909, 9908, 8124, 9449, 606, 6701, 1576, 1396, 4085, 9791, 1715, 2867, 2514, 380, 6965, 526, 8590, 724, 9457, 2092, 1348, 5561, 8595, 1093, 3815, 1986, 496, 5537, 5913, 1150, 8618, 9758, 1048, 7787, 6026, 5387, 5010, 270, 7511, 1065]	51.41	1.9682	4113	0.0000	1
velky,cerveny,kruh,23	[3349, 1847, 3149, 3184, 2924, 4596, 9658, 5125, 5133, 3153, 953, 5755, 2923, 2862, 118, 4522]	50.91	1.9938	4073	0.0018	14
velky,cerveny,kruh,4*6	[7008, 6461, 6574, 4619, 2556, 5835, 3474, 1952, 1415, 2925, 6556, 332, 684, 8667, 3807, 5663]	49.89	1.9025	3991	0.0009	4
velky,cerveny,kruh,42/2	[1457, 9584, 6394, 5209, 4449, 5114, 3378, 9861, 2082, 3367, 2123, 3009, 5842]	58.03	1.8704	4642	0.0017	14
velky,cerveny,kruh,3*7	[9578, 2267, 3739, 5114, 8385, 8891, 273, 6396, 4865, 3619, 1807, 7132, 903, 2945, 9282, 1457]	55.88	2.0020	4470	0.0011	5
velky,cerveny,kruh,23	[3646, 7562, 2529, 3494, 8828, 6598, 8038, 7413, 3103, 562, 6888, 8575]	57.60	1.9426	4608	0.0014	7
velky,oranzovy,kruh,(14+5)	[5941, 6408, 6409, 1456, 1694, 6161, 6200, 7923, 1978, 9325, 5067, 8564, 5765, 9364, 1132, 5566, 1401, 5977, 1240]	48.63	1.9830	3890	1.0000	0
velky,cerveny,kruh,(24-2)	[9876, 2165, 3274, 4929, 3850, 6870, 4717, 5129, 1184, 269, 2021, 3275, 5412, 5395, 6537, 5415, 3091, 4915, 8371]	53.30	1.9399	4264	0.0010	4
velky,cerveny,kruh,2	[1511, 9539, 7079, 2147, 2434, 1502, 4808, 135, 4462, 7556]	49.60	2.1242	3968	1.5000	0
velky,oranzovy,kruh,19	[9305, 6764, 7778, 3549, 5225, 9394, 2484, 1219, 1219, 3137, 4225, 2859, 8310, 1928, 3982, 4427, 7609, 3631, 848]	55.40	1.9103	4432	1.0011	5
velky,oranzovy,kruh,16	[9257, 8303, 9332, 1414, 191, 7067, 2206, 8190, 2303, 3645, 8107, 7992, 4179]	52.74	1.9336	4219	1.0000	0
velky,cerveny,kruh,(9+21-5)	[6750, 2789, 3499, 4464, 5389, 7177, 809, 2413, 4050, 7472, 7282, 7093, 7491, 4313, 7883, 5521, 6425, 8597, 459]	53.69	2.0250	4295	0.0011	5
velky,cerveny,kruh,(16+5)	[6280, 419, 4781, 5371, 6766, 8363, 2339, 4426, 4569, 8910, 4216, 9427, 2655, 1111, 879, 3474, 3010, 9973, 63]	50.65	1.9538	4052	0.0000	0
velky,cerveny,kruh,(37+8)/2	[948, 5507, 444, 6849, 313, 784, 6743, 2933, 8927, 3681, 7656, 5901, 4728, 1877, 2807, 2733, 7582, 9184, 1066]	50.25	1.9613	4020	0.0010	5
velky,oranzovy,kruh,972/57	[7110, 5360, 9053, 7324, 6762, 458, 4931, 7122, 1887, 6691, 3452, 111, 8921, 6404, 8734, 6537, 5647, 9160, 4725]	57.84	1.9271	4627	1.0006	1
velky,cerveny,kruh,24	[8373, 3287, 6139, 7584, 2870, 7518, 3736, 6675, 6944, 817, 6044, 8703, 8547, 6360, 1689, 1181, 6810, 4799, 5891]	49.15	1.9938	3932	0.0013	8
velky,cerveny,kruh,21	[360, 2059, 5609, 3974, 5344, 9514, 6607, 5743, 9271, 9803, 169, 6740, 1723]	50.33	2.0792	4026	0.0000	0
stredni,cerveny,kruh,23	[4815, 8422, 2764, 6509, 8504, 9302, 1207, 1927, 683, 8168]	53.18	1.9539	4254	1.0010	5
velky,cerveny,kruh,24	[6344, 3275, 364, 499, 7976, 1650, 8650, 6253, 3144, 8143]	52.46	1.9419	4197	0.0009	3
velky,cerveny,kruh,(3+21)	[140, 521, 884, 4769, 5818, 5053, 2253, 5042, 238, 5104, 4303, 6053, 1161, 1967, 1103, 6911]	49.58	2.0075	3966	0.0008	2
velky,cerveny,kruh,25	[2012, 2021, 5424, 6679, 7604, 1164, 2161, 4681, 3935, 9392, 5073, 4385, 8262, 2221, 9071, 5432, 4855, 4937, 3747]	53.91	1.9753	4313	0.0009	3
velky,oranzovy,kruh,16	[2667, 5633, 4048, 2269, 7799, 3312, 3609, 2921, 2146, 801]	60.31	1.9475	4825	1.0008	2
velky,cerveny,kruh,95/4	[6606, 7073, 1759, 5129, 516, 6146, 5066, 325, 2145, 579, 296, 7625, 4764, 9473, 9385, 5515, 6742, 4032, 2691]	50.16	1.9536	4013	0.0006	1

3. Sada

velky,cerveny,kruh,(15+6)	[2697, 2294, 4349, 3673, 7448, 4616, 2097, 2087, 6908, 1913, 4573, 1736, 915, 2799, 1522, 5957, 9320, 4300, 4981]	57.79	1.7498	4623	0.0000	0
velky,cerveny,kruh,21	[2592, 6449, 6174, 2014, 4283, 4864, 7777, 7429, 8011, 4333, 3542, 7136, 9447, 1511, 8195, 6920]	47.35	1.5539	3788	0.0009	5
velky,cerveny,kruh,21	[1267, 2054, 8749, 6559, 6674, 5448, 3889, 5431, 2391, 8233, 9925, 8376, 1664, 2818, 7173, 4985, 2745, 2190, 3143]	48.88	1.6800	3910	0.0009	5
velky,cerveny,kruh,23	[6601, 4685, 8119, 7184, 3623, 2682, 8596, 9497, 2883, 1372]	48.08	1.7197	3846	0.0014	13

velky,cerveny,kruh,24	[6717, 6017, 3934, 469, 1919, 3522, 2548, 2477, 4374, 9594]	57.26	1.7307	4581	0.0007	3
velky,cerveny,kruh,23	[7524, 9020, 7634, 1769, 9371, 6156, 6949, 9287, 6353, 6976, 6543, 2512, 2912, 9887, 3183, 324, 5244, 6601, 8757]	53.01	1.7020	4241	0.0007	3
velky,cerveny,kruh,22/1	[1772, 6944, 8874, 8644, 8919, 1806, 3025, 7127, 4092, 9759, 8018, 3213, 2051]	56.85	1.6812	4548	0.0014	11
velky,cerveny,kruh,0	[6151, 3874, 4409, 1220, 2085, 3269, 7860, 760, 2955, 8602]	59.74	1.6751	4779	1.5000	0
velky,cerveny,kruh,(180+23)/9	[5052, 7654, 2012, 7048, 5114, 1827, 431, 3766, 6979, 5608, 5569, 2466, 878, 3595, 6787, 43, 6446, 3661, 1963, 8836, 5360, 7356, 6411, 8159, 2486, 390, 5991, 6576, 7509, 1714, 8644]	54.44	1.5249	4355	0.0000	0
velky,cerveny,sestiuhelnik,24	[5837, 4218, 6434, 1394, 5834, 9160, 2584, 8577, 6284, 8206, 443, 8901, 8980]	50.24	1.4044	4019	1.0000	0
velky,cerveny,kruh,9*(4+8)/5	[8802, 4838, 9979, 5584, 5487, 185, 8339, 5166, 6283, 5225, 624, 6288, 816, 4861, 7328, 5097, 4271, 7105, 8795]	53.48	1.7368	4278	0.0007	2
velky,cerveny,kruh,1	[9216, 604, 9044, 1829, 5519, 9987, 9671, 2545, 9791, 9644]	54.23	1.6249	4338	1.5000	0
velky,cerveny,kruh,(26-2)	[8918, 5873, 2354, 472, 5739, 5282, 1619, 3415, 7381, 3046, 6943, 4913, 602, 7057, 284, 4859, 4805, 8360, 9538]	52.38	1.7823	4190	0.0000	0
velky,cerveny,kruh,3*7	[1419, 4204, 9294, 8531, 2315, 1705, 1359, 9455, 8583, 9307, 7660, 2561, 4240, 8331, 6358, 6131]	50.50	1.7781	4040	0.0000	0
velky,cerveny,kruh,22	[3118, 4475, 3749, 9959, 6248, 2196, 9982, 1957, 5262, 6690, 3993, 8755, 3574, 6707, 7848, 7949, 3425, 6428, 7474]	50.15	1.7587	4012	0.0017	15
velky,cerveny,kruh,22	[1779, 2627, 5804, 5484, 164, 9464, 8326, 329, 7662, 8333, 1120, 6706, 6292, 6833, 12, 7721, 1483, 8946, 6408]	52.44	1.6917	4195	0.0011	9
velky,cerveny,kruh,23	[4164, 1904, 1014, 1749, 6035, 4608, 2836, 2795, 5243, 9939, 8540, 218, 4901, 3189, 5397, 3881, 3635, 4934, 3841]	45.61	1.5543	3649	0.0008	5
velky,cerveny,kruh,98/4	[2998, 2089, 195, 5555, 2454, 3381, 8510, 4221, 8161, 6624, 4708, 2684, 8981, 5425, 6178, 9115, 8377, 5847, 752]	53.16	1.7554	4253	0.0000	0
velky,cerveny,kruh,9*7/3	[6796, 6866, 1204, 1874, 2502, 2553, 9569, 6766, 4356, 2667, 857, 7091, 7694, 4951, 8613, 3892, 4706, 3730, 4819]	48.25	2.0039	3860	0.0008	3
velky,cerveny,kruh,25	[6343, 593, 5184, 8084, 8654, 7620, 2152, 1739, 3835, 1470, 6371, 8933, 9904, 4174, 2259, 6185, 8632, 2789, 9246]	51.00	1.8859	4080	0.0009	4
velky,cerveny,sestiuhelnik,431/20*1	[2071, 3673, 9291, 8085, 3404, 8496, 614, 5316, 6980, 6824, 1970, 7002, 8377, 8251, 3124, 1992, 1281, 1280, 6242, 7467, 5421, 3371, 8283, 1039, 6063, 1962, 1863, 3618, 7748, 4909, 7559]	56.05	1.6964	4484	1.0000	0
stredni,cerveny,kruh,24	[7104, 148, 8094, 2394, 8834, 9054, 532, 2647, 2214, 9121, 6285, 5099, 7217, 3102, 5968, 1010, 2670, 8446, 9345]	61.81	1.7590	4945	1.0005	1
velky,cerveny,kruh,24	[5041, 4817, 9374, 2224, 7181, 628, 8299, 9061, 154, 9615]	58.94	1.5256	4715	0.0010	8
velky,cerveny,kruh,25	[1903, 191, 6719, 3689, 3998, 9998, 280, 9683, 3735, 5773, 2081, 367, 2148, 7533, 611, 5827, 3965, 1528, 3425]	54.76	1.7424	4381	0.0008	4
velky,cerveny,sestiuhelnik,68/3	[5726, 7933, 6131, 2734, 741, 317, 4465, 2703, 5853, 6880, 6863, 6213, 6288, 784, 520, 6950, 1575, 7893, 7305]	54.44	1.6196	4355	1.0000	0
stredni,cerveny,kruh,22	[8322, 8156, 354, 3394, 2354, 7302, 9389, 9042, 8641, 2089, 8970, 9927, 5136]	52.49	1.5769	4199	1.0000	0
velky,cerveny,sestiuhelnik,25	[8067, 9377, 7784, 9658, 9956, 4954, 5860, 2977, 2135, 6668, 3710, 8373, 4672, 2003, 1047, 4225, 4336, 300, 2604]	49.96	1.8654	3997	1.0006	1
velky,cerveny,kruh,24	[7970, 4826, 329, 3369, 8834, 5446, 5599, 1275, 5704, 8556]	56.01	1.7504	4481	0.0006	2
velky,cerveny,kruh,94/4	[4104, 8996, 2674, 8069, 4092, 7236, 1205, 3209, 1094, 2619, 3296, 4013, 9094]	50.69	1.7920	4055	0.0009	5
velky,cerveny,kruh,22	[3136, 7880, 3849, 2639, 2177, 1788, 7849, 1013, 7382, 3646]	47.68	1.5576	3814	0.0013	11
velky,cerveny,kruh,25	[6860, 8366, 779, 7514, 7979, 4706, 9055, 9229, 4765, 4652, 6905, 6351, 6941, 9859, 5318, 6598, 1780, 2649, 5173]	43.50	1.8052	3480	0.0013	9
velky,cerveny,kruh,24	[2716, 173, 2184, 5059, 206, 6268, 397, 5227, 9404, 6226, 8504, 3605, 7586, 7135, 2281, 7392]	51.49	1.7731	4119	0.0007	2
velky,cerveny,kruh,7*3	[8110, 7592, 9834, 8339, 6972, 73, 1527, 6808, 5795, 6467, 2623, 9948, 7188, 2730, 658, 6237, 6639, 6602, 5791]	45.60	1.5668	3648	0.0012	10

stredni,cerveny,kruh,23	[9139, 8020, 224, 7104, 296, 2576, 6328, 8133, 1433, 7304, 5125, 4175, 2256, 6197, 153, 6164, 9709, 9135, 955]	57.79	1.5993	4623	1.0007	3
velky,oranzovy,kruh,4*(7-3)	[7098, 9080, 8033, 9864, 1404, 6581, 2624, 9362, 6739, 8961, 1917, 1544, 2381, 2555, 8663, 5334]	52.98	1.6355	4238	1.0012	7
velky,cerveny,kruh,25	[4255, 2618, 9914, 9874, 6059, 3150, 8461, 3911, 7535, 8771, 2609, 7627, 6439, 4292, 4131, 9940, 1247, 1172, 1312]	56.01	1.6512	4481	0.0006	2
velky,cerveny,kruh,22	[2684, 8180, 4349, 8904, 5156, 6572, 8110, 4647, 3772, 7362]	50.86	1.6201	4069	0.0005	1
velky,cerveny,kruh,6*4	[7993, 8048, 9599, 6949, 708, 1923, 8086, 9934, 1508, 6895, 9034, 4609, 7325, 2487, 795, 6403]	53.41	1.4511	4273	0.0005	1
velky,oranzovy,kruh,(26-7)	[5599, 1421, 7945, 497, 5454, 7113, 253, 2174, 2549, 3223, 7716, 5693, 1337, 2023, 2976, 7146, 5298, 1242, 4006]	57.25	1.7697	4580	1.0000	0
velky,cerveny,sestihelnik,(90-65)	[2560, 8279, 3109, 7513, 9424, 8676, 5984, 7925, 2490, 7961, 8193, 6062, 7871, 987, 8505, 6711]	50.35	1.5746	4028	1.0005	1
velky,cerveny,kruh,22	[7823, 9698, 5769, 3544, 9029, 1932, 8200, 5395, 692, 6717, 2908, 6309, 3430, 4085, 4027, 5043]	50.26	1.7950	4021	0.0010	4
velky,oranzovy,kruh,16	[1682, 392, 1338, 8174, 5699, 5894, 4986, 1133, 8396, 4653]	49.28	1.7717	3942	1.0010	4
velky,oranzovy,kruh,16	[2913, 5567, 433, 1424, 6980, 9510, 9531, 5903, 2856, 1770, 4827, 9590, 4027]	50.01	1.5502	4001	1.0005	1
velky,cerveny,kruh,(23+2)	[9476, 7157, 2614, 7649, 4084, 5018, 370, 7605, 8733, 2693, 7760, 5153, 3382, 3279, 2549, 2740, 8021, 4515, 9607]	52.78	1.6349	4222	0.0010	6
velky,cerveny,kruh,(28-7)	[3996, 5864, 4874, 4074, 1291, 3802, 9091, 4009, 9518, 3845, 47, 5669, 3707, 5694, 8749, 9838]	59.88	1.6185	4790	0.0009	7
velky,cerveny,kruh,8*3	[4994, 2591, 7464, 3934, 3177, 8293, 7428, 2302, 6593, 4521, 1383, 9847, 2699, 5889, 7561, 9884, 9388, 3415, 5617]	50.39	1.7352	4031	0.0008	5
velky,cerveny,kruh,45/2	[5026, 3889, 969, 7109, 1596, 1940, 6083, 2147, 222, 5238, 6331, 5585, 3468, 6186, 6980, 1345, 6088, 3318, 7995]	58.85	1.7309	4708	0.0000	0
velky,cerveny,kruh,21	[4238, 7493, 5989, 3389, 7106, 9900, 7678, 6649, 3281, 1252, 4323, 6921, 6857, 4116, 4372, 2435, 9803, 6901, 3852]	47.03	1.5810	3762	0.0009	4
velky,cerveny,kruh,22/1	[1399, 9144, 2522, 5354, 5268, 5836, 411, 3844, 1571, 2682, 6447, 7293, 1361, 8681, 8980, 571, 2366, 9397, 4652]	49.59	1.4490	3967	0.0000	0
velky,cerveny,kruh,22	[8350, 2471, 624, 2444, 4103, 3270, 4186, 7703, 6812, 6013, 9137, 3840, 3600, 5509, 6945, 9535, 6695, 7932, 1995]	46.73	1.6282	3738	0.0008	4

4. Sada

velky,cerveny,kruh,7	[3951, 8309, 3604, 1889, 1640, 6347, 7937, 6673, 4283, 3134]	44.55	0.4970	3564	1.5	0
velky,cerveny,kruh,5	[6728, 8399, 9008, 3041, 179, 124, 6095, 273, 9254, 1436]	42.20	0.5151	3376	1.5	0
velky,cerveny,kruh,0	[1186, 7748, 9629, 5604, 249, 5855, 30, 2273, 7132, 1038]	45.25	0.5069	3620	1.5	0
velky,cerveny,kruh,8	[2637, 504, 2828, 4114, 3815, 6895, 4588, 5754, 5898, 4074]	42.28	0.4829	3382	1.5	0
velky,oranzovy,kruh,(2+67)/4	[9120, 1498, 2329, 839, 6340, 1045, 5162, 5301, 9246, 2524, 4265, 4298, 2204, 9427, 334, 8903, 2937, 1146, 4043, 1392, 1389, 7945, 1541, 6664, 706]	39.43	0.4683	3154	1	0
velky,cerveny,kruh,75/5	[1462, 7068, 9359, 3998, 650, 5149, 217, 2379, 1165, 7982, 2581, 2679, 1585, 7439, 6533, 477, 216, 7008, 4633]	44.14	0.4832	3531	1.5	0
stredni,cerveny,kruh,(23-2)	[3641, 3184, 7414, 2974, 2124, 3044, 9154, 2753, 9433, 6431, 7385, 9059, 7982, 4881, 5591, 1688, 3072, 2129, 5863]	42.29	0.4750	3383	1	0
velky,cerveny,kruh,6	[3455, 938, 2625, 9206, 9664, 6534, 4868, 4034, 3561, 8734]	45.33	0.4960	3626	1.5	0
velky,cerveny,kruh,8	[3307, 7919, 1394, 8819, 871, 2748, 6384, 8107, 5069, 2746]	44.68	0.4590	3574	1.5	0
velky,cerveny,kruh,(23-2)	[1129, 6764, 2399, 1959, 2941, 7120, 2593, 1154, 4417, 2507, 3319, 4153, 7712, 3368, 6559, 9994, 1716, 9394, 1667]	46.61	0.5322	3729	0	0

velky,cerveny,kruh,8	[2044, 2794, 3062, 4749, 1547, 1917, 4108, 9292, 7951, 2165]	42.34	0.5557	3387	1.5	0
velky,cerveny,kruh,21*1	[182, 7052, 3319, 8044, 495, 6586, 4161, 3996, 4261, 6956, 8397, 5561, 9523, 5703, 8120, 6274, 5784, 2538, 6632]	42.19	0.4880	3375	0	0
velky,cerveny,kruh,6	[2971, 8222, 7414, 5159, 3887, 2189, 9326, 4346, 6656, 7722]	42.91	0.5531	3433	1.5	0
velky,cerveny,kruh,9	[5432, 5174, 7790, 5093, 4303, 6919, 6349, 1875, 6140, 2571]	43.04	0.6595	3443	1.5	0
velky,cerveny,kruh,21	[1172, 698, 2048, 4499, 8374, 9874, 9820, 3799, 2491, 5392, 1310, 6893, 6378]	45.25	0.5413	3620	0	0
velky,cerveny,sestihelnik,(27-5)	[6950, 878, 5134, 1938, 8425, 5144, 6854, 3124, 7189, 5305, 1887, 1081, 7175, 132, 5607, 1088, 3954, 6333, 1629]	44.01	0.5421	3521	1	0
velky,oranzovy,kruh,96/5	[6824, 4074, 2792, 6018, 296, 6167, 944, 974, 1469, 9285, 1084, 1449, 1406, 1271, 7230, 2352, 7970, 5432, 2597]	40.36	0.4827	3229	1	0
velky,cerveny,kruh,(24-2)	[8087, 1924, 5972, 9144, 8137, 9971, 897, 955, 8202, 7284, 9001, 704, 1585, 6934, 7954, 7013, 4411, 7998, 9766]	41.35	0.4961	3308	0	0
velky,cerveny,kruh,0	[8581, 4979, 7985, 3467, 2184, 2733, 7340, 8808, 8224, 9090]	42.95	0.4849	3436	1.5	0
velky,cerveny,kruh,7	[5584, 1164, 8312, 7421, 959, 1307, 8577, 2783, 990, 4920]	47.83	0.5458	3826	1.5	0
velky,cerveny,kruh,18*8	[8704, 2064, 7749, 4559, 2379, 1656, 194, 1405, 6678, 1476, 3558, 1231, 2328, 1879, 5438, 2477, 285, 9813, 7696]	46.46	0.5539	3717	1.5	0
stredni,cerveny,kruh,93/4	[8337, 2404, 9679, 435, 2165, 4040, 7541, 3208, 8130, 834, 3203, 123, 4493, 9737, 6438, 4582, 3888, 946, 8449]	41.33	0.4841	3306	1	0
velky,cerveny,kruh,9	[7693, 7334, 9164, 233, 2364, 9071, 4789, 8706, 752, 8633]	43.15	0.4529	3452	1.5	0
velky,cerveny,kruh,7	[8162, 7244, 6002, 9594, 2396, 2923, 6867, 2464, 2299, 9275]	42.09	0.6130	3367	1.5	0
stredni,cerveny,kruh,24	[6215, 4688, 5294, 1288, 3401, 8464, 4024, 8569, 4484, 9539, 4619, 6927, 4520]	39.73	0.5375	3178	1	0
velky,cerveny,kruh,1	[9284, 4429, 893, 5749, 9746, 1443, 551, 3658, 2294, 3342]	45.21	0.5381	3617	1.5	0
velky,cerveny,kruh,2	[8251, 584, 9304, 9593, 5933, 1687, 1482, 6936, 9782, 4659]	41.93	0.5365	3354	1.5	0
velky,cerveny,kruh,7*3	[9681, 1494, 7472, 3830, 4085, 6855, 9103, 6907, 3249, 6139, 458, 2148, 4918, 3129, 3269, 3408]	37.76	0.4847	3021	0	0
velky,cerveny,kruh,18/5	[3134, 9939, 7609, 180, 8620, 353, 4367, 2835, 7358, 753, 8696, 6581, 7635, 2659, 1412, 8106, 6216, 8784, 1023]	45.53	0.4936	3642	1.5	0
velky,cerveny,kruh,(73+0)	[6034, 1000, 3306, 4183, 8989, 9788, 5136, 8280, 753, 7199, 2815, 7690, 3564, 97, 4334, 5668, 5829, 3781, 4480]	43.00	0.5939	3440	1.5	0
velky,cerveny,sestihelnik,99/4	[1001, 9409, 3257, 4518, 6948, 9242, 9863, 1644, 8003, 879, 4994, 2251, 9074, 3189, 1475, 1580, 1737, 4341, 6112]	40.43	0.4493	3234	1	0
velky,oranzovy,kruh,19	[6383, 374, 4504, 9843, 5078, 4178, 4050, 6677, 3139, 3269, 2377, 1721, 9321]	38.71	0.5326	3097	1	0
velky,cerveny,kruh,4	[3146, 5014, 9974, 9289, 6485, 9651, 8404, 8137, 8457, 3495]	41.33	0.5571	3306	1.5	0
velky,cerveny,sestihelnik,(23-0)	[8336, 1979, 4087, 6014, 9030, 7964, 133, 1665, 5720, 5803, 4438, 5943, 4533, 7497, 5843, 7000, 8338, 4742, 1468]	39.80	0.5384	3184	1	0
velky,cerveny,kruh,74*0	[1892, 4829, 8201, 6663, 2759, 9672, 8774, 4676, 2157, 1349, 8995, 5840, 6634, 5140, 532, 3992, 7667, 7756, 5933]	44.66	0.6016	3573	1.5	0
velky,cerveny,kruh,4	[3050, 7204, 204, 740, 59, 4085, 3004, 7061, 3318, 1928]	41.16	0.4794	3293	1.5	0
velky,cerveny,kruh,9	[3411, 4127, 5574, 6834, 8483, 9877, 229, 2209, 1587, 4364]	42.26	0.5440	3381	1.5	0
stredni,cerveny,kruh,(32-7)	[2128, 3238, 1454, 559, 3110, 9185, 2639, 3639, 4095, 902, 3662, 13, 8947, 1457, 8791, 5311, 7832, 6427, 2637]	40.19	0.5407	3215	1	0
velky,cerveny,sestihelnik,(29-8)	[216, 7912, 3909, 4340, 4566, 262, 6680, 649, 6613, 1339, 4251, 8108, 6078, 2771, 7053, 7128, 1289, 1518, 2245]	43.75	0.4890	3500	1	0

velky,cerveny,kruh,9	[737, 2756, 4763, 7264, 1564, 5047, 3649, 8371, 7276, 841]	44.41	0.4910	3553	1.5	0
velky,cerveny,kruh,9	[4306, 6844, 1444, 545, 2030, 6439, 1779, 1990, 7887, 1537]	48.64	0.4767	3891	1.5	0
velky,cerveny,kruh,8/(6+91+9)	[81, 5433, 9997, 692, 8499, 1147, 4750, 1738, 9, 1975, 1946, 4805, 3732, 7408, 3230, 5435, 4748, 1943, 9301, 2559, 188, 668, 6333, 4963, 2890, 8267, 3194, 8054, 5313, 3490, 286]	44.45	0.5226	3556	1.5	0
velky,cerveny,kruh,(97+3-41-3)	[3499, 5039, 8094, 6044, 7504, 4664, 3469, 3950, 694, 9047, 7442, 8153, 8449, 7852, 3167, 4337, 8587, 8437, 4551, 4271, 1200, 4343, 5931, 5901, 3178, 4548, 5824, 5321, 9012, 9049, 1407, 6636, 435, 6409]	46.60	0.5411	3728	1.5	0
velky,cerveny,kruh,4*(64-8)	[9679, 8831, 3049, 4617, 5915, 6109, 8803, 3908, 5604, 9179, 3433, 7884, 601, 4547, 5688, 9204, 3920, 5074, 3016, 9617, 1011, 1974, 5012, 7640, 6544]	42.98	0.5358	3438	1.5	0
velky,cerveny,kruh,9	[3959, 9611, 7469, 7697, 3025, 849, 9554, 8071, 8007, 6645]	42.03	0.6069	3362	1.5	0
velky,cerveny,kruh,1	[8366, 7004, 41, 6014, 461, 1967, 5891, 3333, 5287, 9894]	43.35	0.4942	3468	1.5	0
velky,cerveny,kruh,6	[8104, 6814, 7433, 8884, 5345, 6761, 5666, 4636, 4931, 4710]	42.10	0.5953	3368	1.5	0
velky,cerveny,kruh,73/3	[6597, 4796, 5309, 9339, 8346, 6423, 3191, 704, 3444, 6717, 9939, 1413, 2273, 3486, 4247, 8351, 4503, 1522, 8]	40.70	0.4701	3256	0	0
velky,cerveny,kruh,7	[862, 8029, 5589, 8399, 9578, 8787, 9177, 7491, 2041, 9366]	44.24	0.5335	3539	1.5	0
velky,cerveny,kruh,(25-1)	[1264, 6234, 3986, 6109, 9594, 9314, 491, 5575, 6886, 6872, 2675, 7541, 5321, 7739, 9694, 6493, 1929, 5307, 4511]	44.31	0.4805	3545	0	0

5. Sada

velky,cerveny,kruh,25	[3305, 5519, 1574, 2474, 548, 8116, 1063, 377, 5255, 9096]	56.59	1.4507	4527	0.0013	14
velky,oranzovy,kruh,17	[6226, 8135, 1898, 1554, 6800, 354, 6264, 2303, 8737, 1720, 6556, 9367, 9947, 1648, 1559, 3954, 3000, 4457, 3383]	56.71	1.2110	4537	1.0008	7
velky,cerveny,sestihelnik,22/1	[4853, 8720, 3267, 5113, 1977, 9228, 1171, 91, 5539, 772, 7406, 7907, 3941, 6514, 3655, 748, 1313, 5777, 6560]	50.58	1.1873	4046	1.0000	0
velky,cerveny,kruh,21	[3068, 4796, 2154, 2069, 8669, 96, 8731, 209, 3851, 6518, 7490, 6101, 8839, 8498, 664, 4524, 1397, 705, 4777]	54.21	1.2982	4337	0.0007	6
velky,cerveny,kruh,(5-56)*9	[276, 3284, 5214, 7333, 2175, 3863, 2810, 2711, 5124, 12, 987, 6445, 4775, 106, 5649, 7895, 1759, 5040, 8285, 5067, 9632, 816, 4441, 1561, 6114]	55.10	1.2400	4408	1.5000	0
velky,cerveny,kruh,26/4	[9170, 584, 7459, 1278, 5912, 5967, 9725, 6347, 3990, 9624, 37, 1595, 9476, 2002, 20, 2689, 3609, 1406, 6722]	54.15	1.3668	4332	1.5000	0
velky,cerveny,sestihelnik,(17+4)	[35, 5479, 3140, 9257, 1644, 2288, 486, 4813, 4884, 9801, 8284, 8553, 6207, 1735, 441, 7184, 5673, 9405, 6853]	55.30	1.2416	4424	1.0000	0
velky,cerveny,kruh,(36+4)	[5650, 714, 2534, 2149, 298, 4782, 5168, 9853, 156, 3053, 7582, 8801, 8704, 6489, 6346, 3041, 7056, 8788, 8014]	54.54	1.3123	4363	1.5000	0
velky,cerveny,sestihelnik,5*5	[6437, 680, 7299, 7313, 9705, 6497, 5165, 1336, 5591, 3267, 7045, 1776, 4922, 3276, 861, 4009, 2668, 3858, 3607]	50.94	1.2933	4075	1.0004	1
stredni,cerveny,kruh,51/2	[3015, 9136, 7920, 1747, 2144, 3953, 9024, 2572, 5368, 7647, 2031, 627, 4772, 2576, 5260, 2286, 8996, 8068, 3775]	49.84	1.3138	3987	1.0000	0
stredni,cerveny,kruh,3*7	[6821, 9940, 8344, 574, 591, 5489, 1323, 3098, 8048, 6137, 827, 6798, 2244, 9919, 5603, 5044, 9966, 6156, 2673]	54.76	1.2190	4381	1.0010	11
velky,oranzovy,kruh,(8-68+79)	[3894, 3134, 9483, 5114, 6595, 2081, 148, 5110, 1911, 3630, 8069, 7295, 1028, 6716, 7394, 7918, 6873, 5957, 5209]	53.13	1.3615	4250	1.0005	2
velky,cerveny,kruh,5	[5156, 7175, 4688, 8237, 3585, 4979, 1989, 3475, 1920, 9842]	56.46	1.3562	4517	1.5000	0
velky,oranzovy,kruh,16	[8240, 608, 7503, 1689, 3392, 334, 3078, 1481, 2656, 8679]	60.09	1.2126	4807	1.0004	1
velky,cerveny,kruh,24	[313, 8378, 3219, 8509, 5648, 5336, 8470, 983, 2014, 1551, 9237, 3544, 2985, 6548, 9784, 5460, 8956, 8977, 7246]	47.26	1.3886	3781	0.0006	3
velky,oranzovy,kruh,(25-8)	[5348, 6473, 4204, 9583, 7157, 907, 2897, 2569, 7492, 1486, 1607, 3205, 5158, 4972, 807, 7267, 6440, 7515, 1548]	51.84	1.3563	4147	1.0000	0

velky,cerveny,kruh,(24-2)	[6753, 9496, 449, 1739, 5194, 8646, 6589, 5438, 6777, 1841, 6484, 8761, 4262, 728, 7751, 8598, 8564, 840, 8575]	52.86	1.2826	4229	0.0000	0
velky,cerveny,kruh,(15+8)	[2116, 5042, 6394, 2084, 4819, 7902, 6607, 9645, 4239, 815, 7011, 7218, 2018, 805, 5779, 9222, 4263, 7645, 1870]	52.41	1.4046	4193	0.0000	0
velky,oranzovy,kruh,(9+8)	[802, 1475, 9818, 1749, 1231, 8219, 2139, 9791, 5838, 7513, 3588, 2524, 3182, 4312, 4526, 3033, 6797, 7853, 2676]	56.68	1.2318	4534	1.0007	7
velky,oranzovy,kruh,20	[5149, 551, 8843, 3304, 5045, 9060, 3673, 9307, 2510, 2186]	49.90	1.4409	3992	1.0007	6
stredni,cerveny,kruh,(29-8)	[4532, 9991, 6354, 9389, 6622, 224, 7625, 1881, 418, 5062, 2387, 2980, 3629, 3672, 5221, 8220, 3164, 3276, 1214]	57.49	1.3326	4599	1.0000	0
velky,cerveny,kruh,24	[4202, 2555, 1804, 4299, 1355, 6156, 4762, 6697, 7094, 5797]	51.91	1.2871	4153	0.0008	6
velky,cerveny,sestihelnik,34*5/7	[5005, 7340, 5809, 5643, 6552, 5626, 5195, 701, 3904, 7614, 7851, 7741, 7795, 8097, 2144, 8957, 947, 1522, 2615]	58.86	1.4126	4709	1.0011	1
velky,cerveny,kruh,(26-5)	[8094, 6659, 8056, 185, 66, 7575, 95, 6454, 8945, 2617, 3036, 2881, 6535, 7638, 7144, 5200, 202, 3495, 9413]	52.98	1.2637	4238	0.0000	0
stredni,cerveny,kruh,(14+8)	[8113, 7154, 4949, 487, 8968, 7618, 2835, 8239, 7654, 4037, 6615, 1360, 838, 9389, 5426, 306, 1085, 2880, 5882]	59.79	1.2799	4783	1.0000	0
velky,cerveny,kruh,24	[1744, 7754, 8224, 6669, 7706, 8548, 9325, 9641, 894, 5718, 6518, 9975, 4043, 2392, 6685, 9126]	53.80	1.3222	4304	0.0008	7
velky,cerveny,kruh,91/4	[1688, 8164, 4316, 1122, 1151, 7373, 8677, 4866, 2554, 5289, 711, 9989, 3629, 7264, 2372, 1752, 3811, 4385, 4330]	50.86	1.3003	4069	0.0000	0
velky,cerveny,kruh,25	[6026, 4289, 8299, 454, 6518, 1738, 2152, 9729, 6455, 4256, 67, 6732, 5027, 1092, 1126, 9912, 7902, 6427, 5581]	54.36	1.3696	4349	0.0005	2
velky,cerveny,kruh,76/1	[9856, 4559, 8397, 4656, 6574, 1529, 407, 7643, 5199, 4109, 4061, 6846, 1531, 7613, 6301, 902, 378, 4987, 3603]	47.90	1.3632	3832	1.5000	0
stredni,cerveny,kruh,(7+4+14-4)	[5709, 2189, 2014, 6964, 6790, 2365, 6133, 4933, 1966, 9124, 6444, 2984, 9936, 4837, 4884, 675, 5167, 2726, 5289, 3154, 6525, 9296, 3264, 389, 3086, 5217, 7281, 7487, 2163, 3971, 9629]	56.15	1.3949	4492	1.0000	0
stredni,cerveny,kruh,24	[7218, 8350, 3039, 6589, 9539, 9336, 2809, 9651, 2804, 4260, 9108, 7411, 5512, 2723, 2861, 3381]	52.90	1.2663	4232	1.0011	11
velky,cerveny,kruh,25	[7206, 710, 1464, 5269, 4307, 1982, 244, 7665, 5525, 8044]	44.81	1.2761	3585	0.0004	1
velky,cerveny,kruh,72/3	[9233, 2909, 5754, 2149, 6962, 9651, 5302, 6347, 2519, 4443, 6335, 6963, 2632, 2427, 6501, 2301, 8979, 3281, 4251]	45.59	1.3157	3647	0.0000	0
velky,cerveny,kruh,25	[42, 899, 1514, 2869, 4418, 6388, 3538, 6009, 2465, 9941]	59.20	1.4406	4736	0.0011	11
velky,cerveny,kruh,24	[2600, 7571, 3099, 7034, 2663, 4408, 5779, 1623, 7694, 4957, 6797, 8303, 7257]	49.99	1.4266	3999	0.0004	1
velky,cerveny,kruh,5	[4480, 2254, 7151, 5925, 1185, 4239, 4826, 2826, 6979, 294]	54.96	1.2576	4397	1.5000	0
velky,oranzovy,kruh,18	[2525, 5528, 4988, 9634, 3800, 1452, 5886, 703, 5118, 9083, 3523, 2183, 3936, 609, 6009, 5806]	52.59	1.3328	4207	1.0004	1
velky,oranzovy,kruh,17	[7837, 2324, 3408, 5929, 6524, 9134, 6129, 2671, 7267, 3545]	50.29	1.2947	4023	1.0009	9
velky,cerveny,kruh,23	[6583, 2849, 7684, 4184, 4139, 6298, 505, 3831, 9013, 9181]	52.15	1.2623	4172	0.0008	9
velky,cerveny,kruh,(21-0)	[7758, 5039, 6449, 3695, 5278, 5570, 6656, 2923, 4171, 1747, 163, 1670, 3061, 6402, 7457, 3619, 1423, 5759, 218]	54.98	1.2979	4398	0.0000	0
velky,cerveny,kruh,(65-4)	[2592, 2659, 5531, 7327, 8469, 62, 8509, 7781, 8835, 1610, 7915, 3744, 1021, 6052, 1087, 7893, 5790, 8507, 7957]	62.55	1.2817	5004	1.5000	0
stredni,cerveny,kruh,23	[2941, 766, 2369, 6924, 5390, 8820, 2989, 2437, 3083, 5666, 1641, 3781, 7832, 9771, 454, 1775, 9855, 9872, 3426]	61.46	1.3206	4917	1.0005	2
velky,oranzovy,kruh,19	[6352, 2939, 9678, 7584, 9626, 5444, 9468, 8253, 1799, 2372]	48.05	1.3689	3844	1.0009	9
velky,cerveny,kruh,(13+6)	[5436, 4550, 6834, 94, 1509, 3857, 2882, 2585, 2126, 5258, 1251, 3775, 5273, 6306, 6559, 3335, 1418, 8901, 8245]	53.59	1.4348	4287	1.5000	0
stredni,cerveny,kruh,25	[995, 3829, 6629, 5004, 9185, 4542, 307, 8923, 9185, 9416, 2942, 3720, 3140, 2538, 4963, 3913]	57.31	1.4519	4585	1.0006	2

velky,cerveny,kruh,(95-74)	[1401, 6767, 8144, 4684, 7258, 3066, 5948, 4913, 995, 7193, 6987, 5080, 4290, 709, 7904, 5879, 998, 3484, 2611]	57.09	1.4951	4567	0.0006	2
velky,cerveny,kruh,(7+18)	[3238, 9878, 2949, 2209, 6244, 5967, 6617, 9659, 1180, 2206, 396, 7483, 7128, 6774, 9736, 9232]	56.76	1.4177	4541	0.0010	10
velky,cerveny,kruh,25	[7674, 1676, 8309, 7654, 5633, 76, 2188, 2985, 8445, 6050]	56.56	1.3803	4525	0.0006	4
velky,cerveny,kruh,2	[9887, 7736, 8959, 6509, 6458, 2453, 2492, 4679, 9324, 9761]	58.03	1.3560	4642	1.5000	0
velky,oranzovy,kruh,19	[1616, 9866, 3683, 7849, 2663, 1348, 8811, 6077, 6989, 277, 1607, 3141, 2666, 8349, 5674, 5825, 30, 1831, 1242]	52.74	1.3192	4219	1.0004	1
velky,cerveny,kruh,25	[3305, 5519, 1574, 2474, 548, 8116, 1063, 377, 5255, 9096]	56.59	1.4507	4527	0.0013	14

6.Sada

velky,cerveny,kruh,21	[3744, 8498, 9779, 3044, 6008, 7872, 307, 8949, 2431, 4414, 7127, 9767, 6281]	70.08	1.4505	5606	0.0010	5
velky,oranzovy,kruh,17	[6158, 899, 6878, 5919, 8258, 3740, 1593, 521, 7907, 2143, 8768]	64.70	2.7545	5176	1.0015	1
velky,cerveny,kruh,23	[4473, 7298, 1094, 3854, 2825, 168, 1441, 929, 9533, 7811, 9867, 3901, 6737, 4737, 1266, 7812, 2157, 4860]	68.44	1.4681	5475	0.0008	4
velky,cerveny,sestihelnik,25*1	[6692, 1128, 5877, 8156, 371, 9329, 2371, 8194, 4136, 4678, 136, 7271, 9195, 5437, 9623, 9773, 7520, 9203, 2429]	67.85	1.6342	5428	1.0000	0
velky,cerveny,kruh,23	[2526, 4958, 5439, 4134, 6533, 1960, 1549, 9293, 5053, 4349, 8767, 6329, 9591, 2916]	65.00	1.4565	5200	0.0007	4
velky,cerveny,kruh,25	[465, 3695, 4944, 5704, 602, 4754, 5689, 3141, 5335, 5520, 5600]	69.36	1.7927	5549	0.0009	4
velky,cerveny,kruh,21	[3477, 4586, 9294, 4064, 7538, 4236, 4618, 6571, 2231, 5679, 8186, 9388, 6286, 4320]	68.84	1.5546	5507	0.0008	2
velky,cerveny,kruh,7*3	[9811, 1832, 9049, 9579, 3747, 3677, 4157, 4240, 4607, 3575, 4933, 1789, 1215, 4460, 4274, 968, 5032, 4839, 4674]	70.28	2.0735	5622	0.0007	1
stredni,cerveny,kruh,24	[6841, 3688, 3914, 9359, 3836, 3100, 6508, 6011, 1754, 8981, 6311, 8187, 6946, 2175, 7923, 1973]	65.23	1.5996	5218	1.0008	4
velky,oranzovy,kruh,19	[2623, 1580, 9563, 2294, 6818, 4806, 1062, 5619, 2639, 8152, 5368, 5108, 8338]	69.28	1.6033	5542	1.0005	1
velky,cerveny,sestihelnik,21	[9321, 9589, 6521, 3836, 2008, 9014, 4839, 1121, 7372, 6574, 2030, 3853, 1069]	71.58	2.4134	5726	1.0000	0
velky,cerveny,kruh,21	[1122, 8681, 8804, 5664, 6425, 3988, 5617, 3331, 6851, 8327, 4055, 4403, 3879, 4035, 4273, 1488, 8878, 1744]	65.20	1.7535	5216	0.0011	4
velky,cerveny,kruh,5	[4145, 1484, 4954, 7178, 9891, 2565, 9917, 1924, 1696, 4075]	65.40	1.7279	5232	1.5000	0
velky,cerveny,sestihelnik,23	[6107, 9146, 4654, 8313, 8147, 9112, 6625, 2249, 4063, 4594, 838, 2436, 5137]	66.09	1.7435	5287	1.0009	5
velky,cerveny,sestihelnik,25	[5254, 7514, 34, 1273, 7703, 578, 9379, 711, 2285, 5881, 7880, 821]	67.06	1.7027	5365	1.0008	3
velky,cerveny,kruh,22	[1840, 7835, 74, 104, 9380, 6614, 5734, 5429, 9032, 7905, 603, 1448, 4514, 236, 6465]	64.99	1.7965	5199	0.0014	9
velky,cerveny,kruh,22	[8636, 1637, 1364, 6329, 4043, 4188, 4987, 5351, 4822, 4022, 7622, 2587, 6414, 6998, 6439, 4944, 9084, 5062, 30]	66.74	1.7549	5339	0.0012	7
velky,oranzovy,kruh,16	[7493, 7721, 4118, 5589, 2504, 6466, 6624, 3575, 4996, 6126, 6525, 9888, 6564, 1637, 5179]	67.09	1.9431	5367	1.0008	2
velky,cerveny,kruh,(6+99)/5	[2883, 9002, 764, 6479, 5653, 4669, 7696, 8832, 1832, 5102, 6461, 5257, 2429, 5393, 4790, 4631, 6025, 2765, 4762, 5531]	64.15	1.5517	5132	0.0008	4
velky,cerveny,kruh,25	[1533, 4277, 4764, 459, 1889, 8364, 2071, 1733, 6715, 9052, 6275, 9321, 6392]	68.55	1.7933	5484	0.0008	2
velky,cerveny,kruh,(6-3)*7	[9089, 3506, 7189, 4199, 7933, 9471, 2126, 183, 293, 4549, 8873, 5224, 7424, 6655, 397, 667, 4465, 84, 5127]	69.63	1.8546	5570	0.0008	2

velky,cerveny,sestiuhelnik,25	[8731, 4568, 6664, 3068, 5903, 1246, 5419, 5093, 1945, 5211, 8319, 3229, 9594, 9670, 5984, 8730, 213, 3631]	62.56	1.8143	5005	1.0005	1
velky,cerveny,kruh,24	[1511, 6386, 534, 4914, 5510, 9240, 5509, 2425, 4694, 5451, 4060, 7238, 3139, 7669, 882, 4526, 2402, 2777]	66.05	1.6823	5284	0.0008	3
velky,oranzovy,kruh,18	[461, 3638, 2078, 109, 4625, 8062, 6624, 703, 3518, 3220, 9770, 4804]	64.73	1.8357	5178	1.0006	1
velky,oranzovy,kruh,153/8	[4850, 7169, 680, 9126, 7713, 1919, 2341, 7805, 4578, 56, 2862, 8023, 8913, 2691, 1913, 8965, 9795, 7652, 3488, 2770, 4888, 4462]	66.46	1.8411	5317	1.0000	0
velky,cerveny,kruh,21	[3710, 7148, 4164, 6754, 6164, 7932, 7228, 1419, 5471, 149, 4718]	64.79	1.6221	5183	0.0009	5
velky,cerveny,kruh,25	[5479, 362, 8629, 5209, 1289, 8740, 1558, 9225, 855]	66.31	1.8871	5305	0.0011	6
velky,cerveny,sestiuhelnik,(27-4)	[3653, 4706, 6584, 2268, 8590, 4666, 1081, 5155, 3637, 8933, 3987, 6871, 6564, 9815, 4444, 6649, 8203]	64.89	1.7744	5191	1.0009	4
velky,cerveny,sestiuhelnik,24	[4685, 9236, 2194, 3803, 7373, 8350, 3727, 1187, 5894, 9908, 8520]	64.78	1.6995	5182	1.0005	1
velky,cerveny,kruh,22	[2909, 8069, 6074, 9494, 7268, 7404, 4771, 1735, 3362, 8707, 8115, 2686]	63.53	2.0047	5082	0.0005	1
velky,oranzovy,kruh,19	[6664, 9524, 3338, 6119, 5999, 3036, 5355, 7837, 5999, 952, 3126]	65.64	1.7306	5251	1.0016	11
velky,cerveny,kruh,(20+1)	[8221, 9686, 8479, 1102, 7318, 3837, 8089, 6390, 8408, 1838, 7858, 9657, 4871, 2710, 981, 6575, 1344, 6080, 519]	63.85	1.9118	5108	0.0000	0
velky,cerveny,kruh,22	[8704, 2306, 6599, 1334, 938, 9014, 901, 6155, 5542, 8978, 9598, 2654]	66.56	1.4766	5325	0.0007	4
velky,cerveny,kruh,(29-8)	[7177, 7579, 3653, 2476, 2648, 6887, 4208, 5627, 7519, 8460, 4177, 137, 1649, 6050, 4617, 4310, 8792, 218, 5207]	67.83	1.4188	5426	0.0000	0
velky,cerveny,sestiuhelnik,(14+9)	[6038, 2689, 8738, 3743, 8816, 949, 1493, 6319, 4767, 5194, 4869, 621, 8796, 8002, 5003, 2679, 511, 3271, 729]	65.90	1.4801	5272	1.0000	0
velky,oranzovy,kruh,17	[4846, 2504, 2393, 5394, 4457, 9636, 8640, 5087, 677, 2065, 7273, 2319, 4690]	67.11	1.5997	5369	1.0009	3
velky,cerveny,kruh,3*8	[9434, 2744, 6674, 8099, 4662, 1477, 4103, 5416, 2651, 3411, 2088, 5419]	66.40	1.6876	5312	0.0008	2
velky,oranzovy,kruh,20	[8130, 2546, 5233, 574, 509, 3600, 2791, 275, 1320, 1548, 7662, 3858, 8369, 1991, 1416, 9569, 9587, 3744, 3098, 3026]	65.50	1.3770	5240	1.0007	4
velky,cerveny,sestiuhelnik,(30-5)	[9170, 8920, 7205, 8593, 9563, 8039, 95, 9344, 1715, 5464, 6107, 857, 460, 3026, 56, 9479, 6519, 7290, 7285]	66.61	1.5651	5329	1.0000	0
stredni,cerveny,kruh,(5+15+1)	[2987, 9519, 4129, 4732, 385, 4342, 5078, 5018, 8474, 5014, 2736, 8363, 8875, 7319, 8535, 2775, 8861, 786, 4113, 4919, 1288, 6920, 3511, 6711, 9842]	66.50	1.6975	5320	1.0000	0
velky,oranzovy,kruh,19	[8298, 3140, 6878, 5969, 9269, 7648, 9531, 6317, 4379, 9613, 6227, 8479, 6088]	65.89	1.6500	5271	1.0004	1
velky,cerveny,sestiuhelnik,23	[3573, 9800, 904, 3003, 1952, 5438, 9478, 2063, 6773, 995, 6748, 3155, 2976, 698, 3202, 6374, 4850, 5802, 1002, 5653]	65.08	1.4095	5206	1.0012	11
velky,oranzovy,kruh,16	[2961, 4739, 5943, 8219, 4745, 7906, 9000, 3941, 1416]	67.83	1.5813	5426	1.0010	6
velky,cerveny,kruh,23	[9593, 4607, 1019, 1559, 7376, 3140, 424, 5677, 763, 2096, 448, 6476, 6328, 7782]	66.20	1.3985	5296	0.0004	10
velky,cerveny,kruh,24	[4390, 6260, 5844, 8709, 4580, 7274, 4537, 301, 6814, 7478]	68.21	1.2311	5457	0.0010	1
velky,cerveny,kruh,7	[5403, 1769, 4444, 8539, 7118, 7425, 8927, 5970, 6653, 7978]	65.93	1.3546	5274	1.5000	0
velky,cerveny,kruh,25	[6606, 6890, 6614, 6729, 9191, 8982, 532, 5793, 15, 9216]	65.26	1.3912	5221	0.0010	8
velky,cerveny,kruh,83356/3550	[3810, 209, 8157, 7568, 4442, 8755, 6174, 7688, 7306, 9949, 1130, 8824, 5712, 3062, 4468, 8375, 5375, 3112, 6527, 8803, 8950, 1356, 9184, 2713, 6540, 3109, 2476, 5250, 256, 6253, 1541, 5200, 9175, 9510, 1785, 737, 1293]	61.25	1.4248	4900	0.0000	0