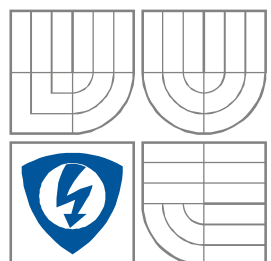


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF BIOMEDICAL ENGINEERING

# OPTIMALIZACE PRO REGISTRACI OBRAZŮ ZALOŽENÁ NA GENETICKÝCH ALGORITMECH

OPTIMIZATION BASED ON GENETIC ALGORITHMS FOR IMAGE REGISTRATION

DIPLOMOVÁ PRÁCE

DIPLOMA THESIS

AUTOR PRÁCE

Bc. PAVLA RONKOVÁ

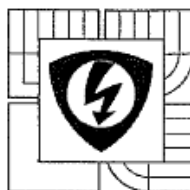
AUTHOR

VEDOUCÍ PRÁCE

Ing. VRATISLAV HARABIŠ

SUPERVISOR

BRNO 2012



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav biomedicínského inženýrství

## Diplomová práce

magisterský navazující studijní obor  
Biomedicínské inženýrství a bioinformatika

**Student:** Bc. Pavla Ronková

**ID:** 106117

**Ročník:** 2

**Akademický rok:** 2011/12

**NÁZEV TÉMATU:**

**Optimalizace pro registraci obrazů založená na genetických algoritmech**

**POKYNY PRO VYPRACOVÁNÍ:**

1) Prostudujte problematiku registrace digitálních obrazů. 2) Proveďte literární rešerši globálních optimalizačních metod, uvažujte jejich využití pro registraci medicínských obrazů. 3) Navrhněte princip optimalizační metody a vytvořte vhodná testovací data pro ověření funkce optimalizace. 4) Navrženou funkci implementujte v programovém prostředí Matlab. Dbejte na stejné principy pro práci s funkcí jako je tomu u funkcí v Global Optimization Toolboxu. 5) Navrženou metodu otestujte na dostatečném množství syntetických dat. Uvažujte různé extrémní případy, které mohou u registrace medicínských dat nastat a negativně ji ovlivnit. 6) Proveďte diskuzi navrženého řešení, diskutujte rovněž i výpočetní náročnost optimalizační metody v porovnání s dostupnými optimalizačními metodami.

**DOPORUČENÁ LITERATURA:**

- [1] SEIXAS, F. L., OCHI, L. S., CONCÍ, A., SAADE, D. M. Image registration using genetic algorithms, Proceedings of the 10th annual conference on Genetic and evolutionary computation (GECCO '08), Maarten Keijzer (Ed.). ACM, New York, NY, USA, 1145-1146.  
[2] WEISE, T. Global Optimization Algorithms – Theory and Application, 2009 published by it-weise.de (self-published), Germany. 2009.

**Termín zadání:** 6.2.2012

**Termín odevzdání:** 18.5.2012

**Vedoucí práce:** Ing. Vratislav Harabiš

**Konzultanti diplomové práce:**

prof. Ing. Ivo Provazník, Ph.D.

*předseda oborové rady*

**UPOZORNĚNÍ:**

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

## **ABSTRAKT**

Diplomová práce se zabývá globálními optimalizačními metodami a jejich využitím při registraci medicínských obrazů. Hlavní cíl spočívá ve vytvoření genetického algoritmu a testování jeho funkčnosti na syntetických datech. Kromě testovacích funkcí a obrazců byl algoritmus podroben i skutečným medicínským snímkům. K tomuto účelu se vytvořilo grafické uživatelské prostředí s volbou parametrů podle aktuální potřeby. Přidáním iterační gradientní metody se algoritmus stal hybridním genetickým algoritmem.

## **ABSTRACT**

Diploma thesis is focused on global optimization methods and their utilization for medical image registration. The main aim is creation of the genetic algorithm and test its functionality on synthetic data. Besides test functions and test figures algorithm was subjected to real medical images. For this purpose was created graphical user interface with choice of parameters according to actual requirement. After adding an iterative gradient method it became of hybrid genetic algorithm.

## **KLÍČOVÁ SLOVA**

Lícování obrazů, kritériální funkce, optimalizace, genetické algoritmy, hybridní algoritmy, testovací funkce, testovací obrazce

## **KEYWORDS**

Image registration, similarity measures, optimization, genetic algorithms, hybrid algorithms, test functions, test figures

**Bibliografická citace práce:**

RONKOVÁ, P. Optimalizace pro registraci obrazů založená na genetických algoritmech. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2012, 80 s., 4 příl., Vedoucí diplomové práce Ing. Vratislav Harabiš.

## Prohlášení

Prohlašuji, že svou diplomovou práci na téma Optimalizace pro registraci obrazů založená na genetických algoritmech jsem vypracovala samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autorka uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této práce jsem neporušila autorská práva třetích osob, zejména jsem nezasáhla nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědoma následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne ...

.....  
podpis autora (autorky)

## Poděkování

Děkuji vedoucímu diplomové práce Ing. Vratislavu Harabišovi za vedení a cenné rady při zpracování mé diplomové práce. Dále také děkuji rodičům a přátelům za trpělivost a psychickou podporu.

V Brně dne ...

.....  
podpis autorky

## Obsah:

Obsah:.....	2
Úvod.....	8
1 Teorie registrace digitálních obrazů.....	9
1.1 Geometrické transformace.....	9
1.2 Interpolace.....	10
1.3 Kriteriaální funkce.....	11
1.4 Problémy registrace digitálního obrazu.....	12
2 Optimalizační metody.....	15
2.1 Klasifikace globálních optimalizačních metod.....	15
3 Genetické algoritmy.....	19
3.1 Teoretický základ genetických algoritmů.....	19
3.2 Podmínky pro použití GA.....	22
3.3 Kódování.....	22
3.4 Ohodnocení individuí.....	23
3.5 Selektce.....	24
3.6 Genetické operátory.....	26
3.7 Ukončující podmínka.....	27
3.8 Paralelní genetické algoritmy (PGA).....	27
3.9 Hybridní genetické algoritmy.....	29
4 Využití globálních optimalizačních metod pro registraci medicínských obrazů.....	31
5 Navržení optimalizační metody.....	34
6 Testovací data.....	35
6.1 Testovací funkce.....	35
6.2 Testovací obrazce.....	38
7 Realizace.....	41
7.1 Popis programu.....	41
7.2 Vytvořený genetický algoritmus.....	43
7.3 Grafické prostředí.....	54
7.4 Dostupné optimalizační metody.....	59
7.5 Výsledky.....	60
8 Diskuse.....	70
Závěr.....	73
Seznam literatury.....	76
Slovník.....	79
Seznam zkratk.....	79
Seznam příloh.....	80

## Seznam obrázků

Obrázek 1 Schéma registrace, upraveno [8].....	9
Obrázek 2 Transformace simplexu: reflexe, extenze, kontrakce a zmenšení na polovinu vzdálenosti W-B.....	17
Obrázek 3 Ruletový mechanismus [44] .....	25
Obrázek 4 Farmářský a migrační model paralelních genetických algoritmů.....	29
Obrázek 5 První De Jongova funkce (vlevo) s vrstevnicovým grafem (vpravo).....	36
Obrázek 6 Resenbrockovo sedlo (vlevo) s vrstevnicovým grafem (vpravo).....	37
Obrázek 7 Rastriginova funkce (vlevo) s vrstevnicovým grafem (vpravo).....	37
Obrázek 8 Ackleyho funkce (vlevo) s vrstevnicovým grafem (vpravo).....	38
Obrázek 9 Testovací obrazec složený z pruhů (vlevo) a jeho maska (vpravo).....	39
Obrázek 10 Graf kritériální funkce (vlevo) a vrstevnice (vpravo) vzájemné informace vybrané části identických testovacích obrazců ve formě pruhů.....	39
Obrázek 11 Testovací obrazec složený z rozmazaných teček (vlevo) a jeho maska (vpravo) ...	40
Obrázek 12 Graf kritériální funkce (vlevo) a vrstevnic (vpravo) vzájemné informace vybrané části identických testovacích obrazců ve formě teček .....	40
Obrázek 13 Graf kritériální funkce (vlevo) a vrstevnice (vpravo) korelačního koeficientu vybrané části identických testovacích obrazců ve formě teček; každý pixel.....	40
Obrázek 14 Vývojový diagram hybridního genetického algoritmu.....	42
Obrázek 15 Graf Rastriginovy funkce v oblasti o rozměrech $x,y \in [-2, 2]$ vlevo; kontury dané funkce vpravo.....	43
Obrázek 16 Ohodnocení chromosomů v populaci po průběhu páté iterace.....	46
Obrázek 17 Graf vývoje kvality populace během pěti iterací .....	46
Obrázek 18 Závěrečné vykreslení; vrchní řada: půdorys testovací funkce, algoritmem vypočtený extrém, skutečný extrém; spodní řada: chromosomy počáteční populace, chromosomy během všech iterací, všechny chromosomy zobrazené na testovací funkci.	49
Obrázek 19 Vytvoření ‚vymaskovaného‘ obrazu [30].....	50
Obrázek 20 Šachovnicový způsob vizualizace výsledku optimalizace dvou identických obrazů; vlevo dokonalá, vpravo nedokonalá s hodnotou horizontálního posunu 11 a vertikálního - 27 pixelů.....	51
Obrázek 21 Zprůměrovaný obraz tvořený referenčním a špatně zregistrovaným s hodnotou horizontálního posunu 11 a vertikálním -27 pixelů .....	52
Obrázek 22 Výsledek starého algoritmu transformace vlevo, výsledek zdokonaleného algoritmu transformace vpravo .....	53
Obrázek 23 Levá část a polovina pravé části hlavního dialogového okna s příkazy a nastavenými počátečními parametry .....	57
Obrázek 24 Pravá polovina hlavního dialogového okna; ve vrchní řadě dva načtené vstupní snímky, ve střední řadě aktuální maska vlevo a slícované obrazy vpravo, ve spodní řadě výpis optimálních hodnot transformace a časové náročnosti průběhu.....	58
Obrázek 25 Registrace testovacích obrazů na falešných extrémech; vlevo transformace plovoucího obrazu o $x=13, y=0$ ; vpravo transformace obrazu o $x=13, y=-10$ .....	71
Obrázek 26 Invertované vymaskované snímky buněčné linie s odlišnou pozicí jedné buňky ..	72

## Seznam tabulek

Tabulka 1 Časová náročnost metod selekce.....	14
Tabulka 2 Převod binárního kódu na Grayův kód .....	23

Tabulka 3 Řešení složeného výroku $(A \vee B) \wedge C$ .....	48
Tabulka 4 Rastriginova funkce; prohledává se prostor $x,y \in \langle -2, 2 \rangle$ , extrém se nachází v $x,y=0$ a má hodnotu 0 .....	61
Tabulka 5 Rosenbrockova funkce; ; prohledává se prostor $x,y \in \langle -2, 2 \rangle$ , extrém se nachází v $x,y=1$ a má hodnotu 0 .....	62
Tabulka 6 Ackleyho funkce; prohledává se prostor $x,y \in \langle -2, 2 \rangle$ , extrém se nachází v $x,y=0$ a má hodnotu 0 .....	63
Tabulka 7 Ackleyho funkce upravená; prohledává se prostor $x,y \in \langle -50, 150 \rangle$ , extrém se nachází v $x,y=0$ a má hodnotu 0 .....	64
Tabulka 8 Funkce podobna medicínským snímkům; prohledává se prostor $x,y \in \langle -100, 100 \rangle$ , extrém se nachází v $x,y=3$ a má hodnotu -0,9416 .....	65
Tabulka 9 Statistické ohodnocení algoritmu HybGa s volbou různých parametrů při použití testovacích obrazců .....	66
Tabulka 10 Legenda pro nastavení parametrů (NP) při použití testovacích obrazců (T-tečky, P-pruhy) vzorkování je vždy 100% .....	67
Tabulka 11 Legenda pro nastavení parametrů (NP) při použití snímku sítnice; vzorkování je vždy 20% .....	67
Tabulka 12 Statistické ohodnocení algoritmu HybGa s volbou různých parametrů při optimalizaci snímku sítnice .....	68
Tabulka 13 Statistické ohodnocení algoritmu HybGa při optimalizaci snímku sítnice bez ohledu na zvolené parametry .....	68
Tabulka 14 Legenda pro nastavení parametrů (NP) při použití snímku buněk; vzorkování je vždy 50% .....	69
Tabulka 15 Statistické ohodnocení algoritmu HybGa s volbou různých parametrů při optimalizaci dvou mírně odlišných snímků buněk .....	69



## Úvod

Optimalizace patří mezi široce využívané procesy v pestré škále aplikací. Aniž si to uvědomujeme, každou chvíli přemýšlíme nad optimálním rozhodnutím. Zda si koupit kvalitní oblečení s určitou jistotou záruky, či méně kvalitní, které je ovšem levnější a v některých z mála případů může i předčít naše očekávání. Zda si koupit horské kolo nebo silniční. Zda jít do kina či do divadla. Vymyslela by se spousta dalších příkladů, z běžného života. Avšak optimalizace v sobě skrývá velice potřebný nástroj i v případech, které řeší kvalifikovaní odborníci. Ekonomové řeší optimalizace obchodu, strategického řízení, nákupu, propagace za účelem vyššího zisku. S tím souvisí i logistika, která má za cíl efektivní řízení logistických toků. Jedním příkladem z obrovského množství optimalizací v technice je optimalizace výkonu motoru za účelem získání maximálního výkonu při minimální spotřebě. Nemalý význam nabývá tato metoda v medicíně jak v oblasti diagnostiky, tak terapie. Při diagnostice ji lze využít k registraci obrazů, a tím zjištění informací či změn v organismu, což pomáhá v rozhodování lékařů. Při jakémkoli použití však musí být stanoveny ohodnocené parametry zájmu pro optimalizaci a upřednostňující informace.

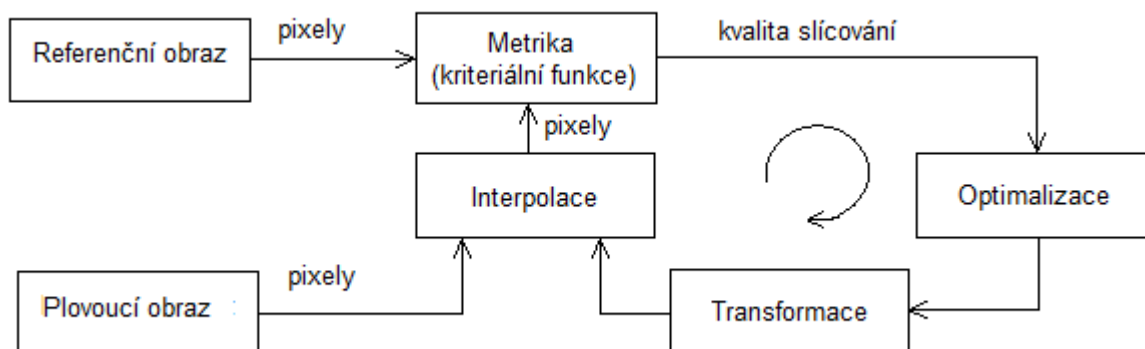
Prvním hlavním cílem této práce je prostudování problematiky registrace digitálních obrazů. Zmíní se i samotná registrace obrazů s vybranými kritériálními funkcemi. V druhé části je ohniskem zájmu přehled optimalizačních metod. Třetí část se zaměří na genetické algoritmy jak po stránce teoretické, tak po stránce využití algoritmů při registraci obrazů. Ve čtvrté kapitole je vytvořen přehled využití globálních optimalizačních metod pro registraci medicínských obrazů. Pátá kapitola obsahuje navržený genetický algoritmus, který se bude implementovat do programového prostředí Matlab. Vytvořené testovací funkce a testovací obrazce pro ověření kvality optimalizace jsou náplní šesté kapitoly. Během sedmé kapitoly je důkladně popsána implementace navrženého algoritmu a na syntetických i reálných datech vyzkoušena jeho funkčnost. Pro práci s obrázky je vytvořeno grafické uživatelské prostředí se snadnou volbou potřebných parametrů optimalizace. V závěru práce je provedena diskuse dosažených výsledků.

# 1 Teorie registrace digitálních obrazů

Registrace neboli lícování obrazů je proces, ve kterém se hledá vhodná geometrická transformace, která obrazy převede do jednoho souřadného systému. Tato transformace tedy popisuje vzájemný prostorový vztah mezi referenčním a registrovaným (plovoucím) souborem dat. Pro matematické vyjádření se využívá vztah [20]:

$$T' = \arg \min_T F(\vec{u}(\cdot), \vec{v}(T(\cdot))) \quad (1)$$

kde  $T$  značí geometrickou transformaci aplikovanou na registrovaný datový soubor,  $T'$  hledanou geometrickou transformaci,  $F$  funkci, která hodnotí registraci dvou datových souborů s danou transformací,  $u(\cdot)$  hodnoty referenčního obrazu a  $v(T(\cdot))$  hodnoty registrovaného obrazu transformovaného do souřadného systému referenčního obrazu.



Obrázek 1 Schéma registrace, upraveno [8]

## 1.1 Geometrické transformace

Geometrická transformace je obecně zobrazení, v němž dochází k rovnosti dvou podmnožin  $A=B$ , kdy každému bodu  $X$  z množiny  $A$  je jednoznačně přiřazen bod  $X'$  z množiny  $B$ . Podmnožina  $A$  je referenčním souborem dat a množina  $B$  registrovaným. Geometrické transformaci můžou podléhat jak body v rámci jednoho souřadného systému, tak samotný systém souřadnic. Transformace souřadného systému se používá při registraci dat z různých souřadných systémů, které pro další zpracování se musí přetransformovat do jednoho.

### Typy transformací

Existují různé druhy transformací, které mohou způsobit translaci, rotaci, změnu měřítka, zkosení atd. Dále se více rozebere translace a rotace [5].

*Translace neboli posunutí* - Při translaci se změní pouze souřadnice objektu, ale vzdálenosti bodů objektu a jejich rovnoběžnost s původním umístěním je zachována. Tato transformace je dána pouze vektorem posunutí  $p=(x_t, y_t)$ .

*Rotace neboli otočení* - Rotace objektu v soustavě souřadnic změní hodnoty souřadnic, ale zachová se vzdálenost bodů objektů. Na rozdíl od translace se však nezachovává rovnoběžnost posunutí bodů.

## **Charakter transformace**

*Rigidní transformace* - Jedná se o afinní transformaci, která se skládá z rotace a posunu. Zápis obecným vzorcem je  $A = TR$ , kde  $T$  značí matici translace a  $R$  matici rotace. Pro přesné určení transformace v dvojrozměrném prostoru potřebujeme znát 3 parametry, kterými jsou posun v ose  $x$ , posun v ose  $y$  a rotace o daný úhel. V trojrozměrném prostoru se počet parametrů zvyšuje o tři.

*Podobnost* – Přidáním dalšího parametru  $s$  k rigidní transformaci získáme afinní transformaci zachovávající úhly a rovnoběžnost přímek. Parametr  $s$  tedy určuje uniformní změnu měřítka. Obecný vzorec bude tedy ve tvaru  $A = TSR$ , kde  $S$  je matice změny měřítka.

*Obecná afinní transformace* – Obecný předpis dané transformace je ve tvaru  $A = TSHR$ , kde  $H$  značí zkosení. Přitom se však předpokládá odlišná změna měřítka ve směrech každé z os, čímž počet parametrů ve dvourozměrném prostoru se zvýší na sedm a pro třírozměrný prostor na dvanáct (tři posuny ve směru každé z os, tři rotace, tři hodnoty zkosení a tři různá měřítka zvětšení).

*Projektivní transformace* – Je transformací, která nespĺňuje podmínky afinity a tudíž nespĺdá do metod lineární registrace. Zachovává zobrazování přímek na přímky, avšak bez zachování jejich rovnoběžnosti.

*Elastická transformace* – Bývá označována též jako deformační pružná nebo obecně nelineární transformace. Výstupem zobrazování přímek jsou křivky.[8]

## **1.2 Interpolace**

Registrace a tedy i transformace dat se neobejde bez jejich interpolace. Původní obraz je v diskrétní podobě. Jeho transformace může způsobit ztrátu informace při přesunu jasové hodnoty do místa mimo vzorkování. Proto se musí počítat s provedením nějakého typu interpolace, která převede jasové hodnoty do míst zobrazení. Pro názornost si představme mřížku, jejíž uzly odpovídají celočíselným hodnotám, takže pokud se kvůli transformaci přesune obraz z míst uzlů do míst prázdných čtverců, tak se hodnota nezaznamená. [3]

Existují různé typy interpolací, jejichž daný výběr závisí na upřednostnění rychlosti, nebo přesnosti výpočtu. Mezi základní typy patří interpolace metodou nejbližšího souseda nebo bilineární interpolace [5].

### 1.3 Kriteriaální funkce

Daný předpis porovnává všechny možné případy a ohodnocuje míru podobnosti mezi referenčním a registrovaným obrazem. Může k tomu využívat následujících metod.

#### Suma rozdílů čtverců (SSD – Sum of Squared Differences)

Nejjednodušší metodou je porovnání referenčního a registrovaného obrazu tím, že je od sebe odečteme. Pokud jsou tyto obrazy shodné, je suma kvadrátů intenzit všech jejich bodů rovna nule (minimum). Metoda se používá však jen u obrazů, u kterých se předpokládá maximální shoda a žádný šum, protože již při malých odlišnostech může metoda selhávat. Větší shody a odstranění šumu se může dosáhnout i předzpracováním obrazů.

Kriteriaální funkce, jejíž minimum hledáme, je ve tvaru [5]:

$$SSD = \frac{1}{N} \sum_i^N |R(i) - L'(i)|^2 \quad (2)$$

pro všechna  $i \in R \cap L'$ , kde N je počet shodných bodů, i je příslušný bod, R je vstupní obraz (referenční) a  $L'$  je zpětně transformovaný obraz (registrovaný).

#### Normovaný korelační koeficient

Korelační metoda může být použita pro zjištění míry podobnosti mezi dvěma obrazy. Její matematická definice je [5]:

$$CC_{(R,L)} = \frac{\sum_i [R(i) - \bar{R}][L'(i) - \bar{L}']}{\sqrt{\sum_i [R(i) - \bar{R}]^2 \sum_i [L'(i) - \bar{L}']^2}} \quad (3)$$

pro všechna  $i \in R \cap L'$ , kde R a  $L'$  představují střední hodnoty intenzit bodů obrazů.

Pokud mají intensity obrazů lineární vztah, pak použití korelačních koeficientů se bere za nejlepší možnost. Funkce nabývá hodnot v rozmezí  $\langle -1; 1 \rangle$  [7].

#### Vzájemná informace (MI – Mutual Information)

MI odhaduje obecnou (tedy i nelineární) závislost dvou souborů dat. Tato metoda je tedy robustnější a vhodná i pro registraci dat získaných z různých zobrazovacích modalit. Je proto nezbytná pro řešení zvláště lékařských obrazů, kdy chceme porovnávat obrázek s anatomii a obrázek s funkcí dané oblasti těla.

Výpočet MI je založen na zjištění statistické závislosti mezi dvěma datovými soubory.

Metoda předpokládá, že obrazy (datové soubory) jsou náhodnými veličinami, a počítá s mezními pravděpodobnostmi výskytu jasových hodnot v jednotlivých obrazech  $p_R[R(i)]$  a  $p_{L'}[L'(i)]$  a se sdruženou pravděpodobností  $p_{RL'}[R(i)L'(i)]$ . Pomocí vzorce s těmito hustotami pravděpodobností se zjistí statistická vazba registrovaného obrazu k referenčnímu [20]:

$$MI(R, L') = \sum_{i \in R} p_{RL'}(R(i), L'(i)) \log_2 \frac{p_{RL'}(R(i), L'(i))}{p_R(R(i))p_{L'}(L'(i))} \quad (4)$$

jehož maximem je dána optimální registrace. Odhad společné i mezních pravděpodobností se může spočítat jako odpovídající relativní četnosti – tedy jako normované histogramy intenzit obrazů  $H_R$ ,  $H_{L'}$ , a normovaný společný histogram intenzit  $H_{RL'}$ . Ze sdruženého histogramu  $H_{RL'}$  lze získat odhady jednotlivých pravděpodobností pomocí vzorce [20]:

$$p_{RL'}[R(i)L'(i)] = \frac{H[R(i)L'(i)]}{\sum_{R, L'} H[R(i)L'(i)]} \quad (5)$$

MI lze vypočítat i pomocí entropií, kdy MI mezi dvěma hodnotami R a L' je dána vztahem [5]:

$$MI(R, L') = H(R) + H(L') - H(R, L') \quad (6)$$

kde H reprezentuje entropii.

## 1.4 Problémy registrace digitálního obrazu

Digitální obraz je diskretní reprezentací informace dvourozměrného prostoru. Pokud je rastrového typu, je popsán pomocí konečného počtu pixelů uspořádaných do mřížky s konečným množstvím barev.

Každý obraz, který má být výpočetně zpracován musí být ve formě diskretních vzorků a každý vzorek musí být reprezentován užitím konečného čísla bitů. To umožňuje informace o obraze zavést do přístroje, který mnohonásobně urychlí jejich zpracování. První operace prováděná při přechodu spojitého obrazu k jeho digitální verzi se nazývá vzorkování a následující operace kvantování.

*Vzorkováním* se hodnoty spojité dvourozměrné funkce zaznamenají pouze v předem zadaných intervalech. Proveďte se to vynásobením spojitého obrazu s vzorkovací funkcí, která přes celý svůj prostor obsahuje Diracovy impulsy navzájem od sebe vzdálené o velikost  $\Delta x$  a  $\Delta y$ . Reprezentace obrazu vzorkovaného v ekvidistantní pravoúhlé síti bude ve tvaru [15]

$$f_s(x, y) = f(x, y) \cdot s(x, y) = \sum_i \sum_k f(i\Delta x, k\Delta y) \cdot \delta(x - i\Delta x, y - k\Delta y) \quad (7)$$

kde  $f_s(x, y)$  představuje vzorkovaný obraz,  $f(x, y)$  spojité obraz a  $s(x, y)$  vzorkovací funkci.

*Kvantováním* se rozumí rozdělení oboru hodnot funkce na intervaly, kterým je přiřazena celočíselná hodnota. Podle způsobu rozdělení kvantované veličiny se hovoří o kvantování uniformním a neuniformním. Uniformní kvantování využívá konstantní délku kroku na oboru hodnot, zatímco u neuniformního je délka kroku proměnlivá. Pro snadnější realizovatelnost se využívá častěji uniformní kvantování. Výběr zástupné hodnoty pro daný interval se volí v závislosti na použité aplikaci a jejich cílech. Nejfrekventovanější je průměrování celého intervalu, jindy se využije medián či průměr z okrajů.[43]

Vzorkování a kvantování s sebou nesou jeden z hlavních problémů registrace digitálních obrazů, jak je to nastíněno ve slovech německého režiséra Wima Wenderse:

*„ S digitálním obrazem je vztah mezi obrazem a skutečností navždy přerušen. Blížíme se době, kdy nikdo nebude moci říct, zda obraz je nebo není pravdivý“.*

Při vzorkování dochází ke ztrátě informace, neboť nelze zaznamenat nekonečné množství hodnot spojité funkce konečným počtem vzorků. Hodnota v daném vzorku je závislá na kvantování, při kterém dochází ke ztrátě informace také, neboť hodnoty v intervalu jsou nahrazeny zástupnou hodnotou. Tato ztráta se nazývá kvantizační chyba nebo také kvantový šum.

Pro transformaci s využitím translace o celočíselný počet pixelů či voxelů nemusí být použita žádná úprava. Problémy však nastanou při zavedení rotace či non-rigidní transformace, kdy vzorky dvou obrazů nemusí ležet přímo na vzorkovaných pozicích druhého obrazu. Řešením je již zmíněná interpolace. Je třeba si však uvědomit, že se kvůli interpolaci změny hodnoty obrazu na daných pozicích, čímž se do obrazu zanáší určitá chyba. Proto by mohlo u dvou identických obrazů dojít k tomu, že nikdy nedosáhnou předem zadané extrémní hodnoty pro danou kriteriální funkci.

Časová náročnost registrace závisí na mnoha parametrech v různých krocích výpočtu a omezeními či výběrem dané možnosti lze dosáhnout přiměřené časové náročnosti. Obrazy

s velkým množstvím pixelů, například medicínské obrazy, se upravují redukcí počtu pixelů až z desítek tisíc na stovky [10], což vede k velkému ušetření času. Logicky lze odvodit, že i počet transformačních parametrů úměrně prodlouží čas výpočtu registrace. Pokud se bude brát při transformaci v úvahu pouze posun v jedné ose, časová náročnost bude menší než při zvýšení počtu operací o zkosení, rotaci a posun v ose druhé. Rotace o úhel jiný než jsou násobky 90° s sebou přináší prodloužení časového výpočtu i z důvodu interpolace. Zvolení optimalizační metody a zvláště jejich parametrů v sobě zahrnuje obrovské množství rozdílné časové náročnosti. Jelikož se tato práce bude zabývat genetickými algoritmy, můžeme zmínit rozdílnou časovou náročnost při volbě selekce [9]. Nejlepší metoda selekce má časovou náročnost  $O(n)$ , kde  $n$  označuje velikost populace. Mezi metody s takovou časovou náročností patří soubojová selekce, která se snadno používá u paralelních implementací algoritmů, což je v nynější zrychlující se době její obrovská výhoda. Některé další metody zahrnuje Tabulka 1.

**Tabulka 1 Časová náročnost metod selekce**

Metoda selekce	Časová náročnost
Ruletový mechanismus	$O(n^2)$
Stochastické univerzální vzorkování	$O(n)$
Turnajový výběr	$O(n)$
Pořadová selekce	$O(n \cdot \log n)$

## 2 Optimalizační metody

Mezi optimalizační metody patří velké množství algoritmických postupů, které se snaží pro danou situaci vyhledat správné řešení co nejefektivněji. Cílem je nalezení globálního extrému (maxima nebo minima) v co nejkratší době. Pro jednoduché funkce lze výsledné optimum nalézt přímo, avšak ve většině případů obsahují funkce více lokálních extrémů. Na tyto komplikovanější funkce je zapotřebí aplikovat optimalizační metody. Pod pojmem globální optimalizační metoda se tedy rozumí nástroj, který pomáhá nalézt globální extrém v celém prohledávaném prostoru. Obecný postup pro hledání globálního extrému touto metodou je následující [34]:

- určí se několik prvních zkušebních bodů z prohledávané oblasti
- pro vybrané zkušební body se testuje cílová funkce  $f$
- určí se nové zkušební body pomocí optimalizační metody
- poté se cyklus testování a určování nových zkušebních bodů opakuje až do nalezení extrému cílové funkce  $f$ .

### 2.1 Klasifikace globálních optimalizačních metod

Při nejčastějším rozdělení globálních optimalizačních metod se vytváří skupina deterministických metod a skupina metod stochastických. Deterministické metody se využívají, pokud je zaručeno nalezení správného řešení. Jelikož jsou určeny známé parametry a do metody se nezanáší žádné pravděpodobnosti, výsledek je vždy za konstantních podmínek invariantní. V případě, že vztah mezi řešením a ohodnocením funkce není zcela zřejmý, nebo je komplikovaný, či prohledávaný prostor je příliš velký, využívá se metod stochastických. Stochastické algoritmy prohledávají heuristicky celý prostor. Heuristikou se rozumí postup, v němž se využije náhoda, zkušenost, intuice. Rozdíl mezi deterministickým a stochastickým algoritmem bude tedy v tom, že na rozdíl od deterministického algoritmu stochastický nezajišťuje nalezení ideálního řešení, ale nabízí nalezení optimálního řešení v optimálním čase. [36]

Spojením těchto dvou hlavních metod uměle vzniká skupina hybridních metod. Podle daného problému začínají nalezením suboptimálního řešení pomocí stochastického algoritmu a následně je aplikován deterministický algoritmus pro nalezení přesného řešení.

Jelikož práce se zabývá genetickými algoritmy, některé další metody se zde pouze nastíní z důvodu prezentace široké škály počátečních myšlenek.

#### Prohledávání hrubou silou

Jedním z přístupů optimalizace je úplné prohledávání neboli prohledávání hrubou silou. Metoda se snaží určit všechna možná řešení a díky nim se rozhodnout pro nejlepší. Při



prohledávání celého prostoru je tedy stoprocentně zaručeno nalezení ideálního řešení, avšak metoda má svá omezení. Je proveditelná jen pro malý počet parametrů a pro malý prohledávaný prostor, neboť s rostoucím počtem parametrů a velikostí dat exponenciálně roste výpočetní náročnost algoritmu [25].

### **Metoda největšího spádu**

Do velké skupiny gradientních optimalizačních metod se zařazují deterministické iterační algoritmy, ve kterých přírůstek vektoru parametrů je úměrný gradientu účelové funkce [16]. Je tedy vyžadována diferencovatelnost účelové funkce. Tyto metody jsou vhodné jen pro řešení určité skupiny problémů, ve kterých je například požadavek konvexního tvaru cílové funkce, což zamezí uvíznutí v lokálním minimu, či dostatečně velká zastavovací podmínka pro dosažení extrému. Nejjednodušší gradientní metodou je metoda největšího spádu. Její podstata spočívá ve vyhledání minima funkce ve směru jejího největšího spádu, který se vypočítá pomocí záporné hodnoty gradientu. Tento nový bod je zvolen výchozím bodem pro další krok, ze kterého se opět volí směr největšího spádu. [37]

### **Nelder-Mead metoda**

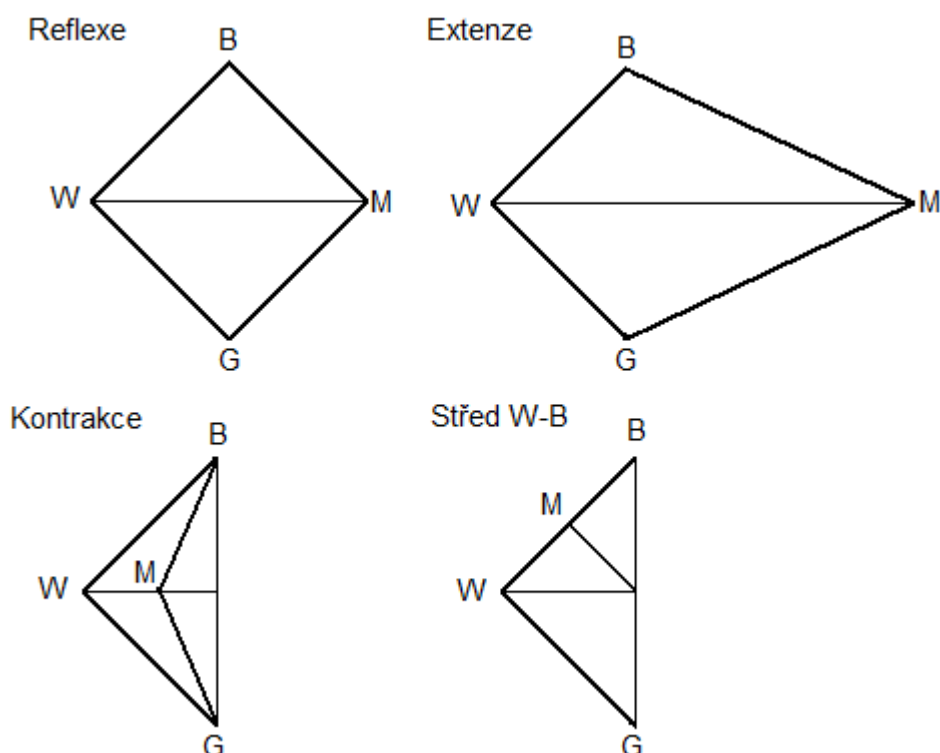
Nejznámější deterministickou negradientní metodou je Nelder-Mead metoda [NM]. Je nazývána také metodou simplexovou, neboť pro vyhledání extrému funkce využívá geometrických útvarů - simplexů. Tvar simplexu je závislý na počtu rozměrů prostoru, neboť je tvořen v  $N$ -rozměrném prostoru  $N+1$  body. V dvojrozměrném prostoru se tedy bude jednat o rovnostranný trojúhelník, v třírozměrném prostoru o pravidelný tetraedr. Při určování nově testovaných bodů se algoritmus řídí dvěma pravidly. Prvním je postup, kde bod s nejhorší funkční hodnotou v simplexu se vypustí a nahradí se novým vrcholem tak, aby opět vznikl rovnostranný trojúhelník. Druhým pravidlem je zákaz navracení se do právě vypuštěného bodu. Pro pracování v dvojrozměrném prostoru se na počátku vygenerují tři body, které jsou vrcholy trojúhelníka. Pomocí ohodnocující funkce se zjistí, který bod je pro optimální řešení nejlepší (B), dobrý (G) a který nejhorší (W). Stanoví se střední bod (M) přesně v polovině vzdálenosti B a G a určí se nový bod (R), pro který se bude zkoumat ohodnocení. Bod R se určí na polopřímce z bodu W do M postupně reflexí, extenzí (dvojnásobnou reflexí), kontrakcí (polovinou délky reflexe z bodu M) nebo středem vzdálenosti bodu W a B jak je tomu na Obrázku 2. Bod R se tedy bude nacházet na místě, na kterém získá větší ohodnocení, než bod W. Situace se opakuje do doby, kdy již nedochází k dalším zlepšením. Nevýhodou je podobně jako u ostatních geometrických metod vysoká pravděpodobnost uvíznutí v lokálním extrému. [36, 41]

### **Slepý a horolezecký algoritmus**

Z velkého množství stochastických metod se zmíní základní stochastické algoritmy (slepý algoritmus a horolezecký algoritmus). Slepý algoritmus opakovaně generuje

náhodné řešení z prohledávané oblasti a zapamatovává si ho jen tehdy, pokud je lepší, než řešení předchozí. Pojem ‚slepý‘ se odvozuje od faktu, že neobsahuje žádnou strategii konstrukce řešení z předchozích řešení [21].

Horolezecký algoritmus prohledává okolí zvoleného bodu a jako další bod vybere řešení s nejlepším ohodnocením. Jelikož ukončovací podmínkou je dosažení předem zvoleného počtu iterací, nově zvolený bod v okolí může mít horší ohodnocení než aktuální bod. Existuje však spousta modifikací horolezeckého algoritmu, kdy lze volit velikost okolí či počet generovaných bodů s prohledávaným okolím.



**Obrázek 2 Transformace simplexu: reflexe, extenze, kontrakce a zmenšení na polovinu vzdálenosti W-B**

### Metoda simulovaného žihání

Metoda simulovaného žihání (SIMSA, SA) patří mezi stochastické optimalizační algoritmy, které mají svůj základ ve fyzice. Žihání je proces, při kterém je těleso umístěné do pece vyhřáté na vysokou teplotu a pomalým snižováním teploty se odstraňují vnitřní defekty tělesa. Postupným snižováním teploty se totiž všechny částice tělesa mohou dostat do rovnovážné polohy, a tím zamezit vytvoření metastabilních struktur, které jsou vzdálené od mřížkové struktury s nejnižší energií. Obecně platí, že čím je snižování teploty u algoritmu pomalejší, tím kvalitnější jsou nalezená optima. Podrobnější fyzikální přístup metody lze nalézt v [21]. Algoritmus v každém kroku vybírá ze svého okolí nejlepší řešení, čímž směřuje nejpřímější cestou do lokálního extrému. S malou pravděpodobností však může akceptovat i aktuální horší řešení, čímž se vymaní z uvíznutí v lokálním extrému. [1]

V práci [1] se autoři zabývají porovnáváním metod simulovaného žihání a genetických algoritmů. K registraci využívají datové soubory medicínských obrazů nesoucí informaci o anatomii nebo funkci orgánů. Datové soubory jsou tedy vytvořené různými modalitami. Jako hodnotící funkci vybrali často využívanou vzájemnou informaci. Pro tři různá kódování při použití genetických algoritmů a dvě kódování u simulovaného žihání došli k výsledkům, že nejrychlejší registrace bude probíhat vždy u simulovaného žihání v závislosti na rostoucí velikosti obrazových dat.

## **Rojení částic**

Rojení částic (PSO) je na populaci založená optimalizační technika, která je inspirovaná sociálním chováním živočišných společenstev, jakým je například letící ptačí hejno či plovoucí hejno ryb za potravou. Každý pták je samostatnou jednotkou pohybující se vlastním směrem a rychlostí. Přitom se však zdržuje v oblasti výskytu jiných ptáků, kteří mohou být v hledání potravy úspěšnější. Inicializace systému začíná v náhodném vygenerování počáteční populace podobně jako u genetických algoritmů, avšak na rozdíl od nich PSO nepracuje s žádnými genetickými operátory, jakými jsou křížení či mutace. Průběh zlepšení se děje podél trajektorie ve směru nejlépe ohodnoceného jedince z aktuálního hejna. Všichni jedinci mají pozici (souřadnici) ve vyhledávaném prostoru, náhodně vygenerovaný vektor rychlosti, který udává směr jejich příštího letu, a pamatují si svou dosud nejlepší pozici vypočítanou pomocí účelové funkce. Jedinec s nejlepším ohodnocením uloží svou současnou pozici do společné paměti celého hejna, takže každé individuum ví, kde se nejlepší řešení nachází. Zároveň každý jedinec zkoumá, zda jeho současná pozice je lepší než předchozí, a pokud ano, tak si ji uloží do své paměti. Jakmile každá částice zná nejlepší polohu celého hejna i svou vlastní nejlepší, upraví si svou rychlost a pozici pro další hledání. Při individuálním způsobu chování budou pokračovat svou vlastní cestou, při konzervativním se vrátí na svou dosud nejlepší pozici, nebo pokud podlehnou přizpůsobivému způsobu hledání řešení, tak budou následovat jedince, který zatím našel nejlepší řešení. Výsledné řešení pro každého jedince však ve většině případů bude s určitou pravděpodobností ovlivněno všemi třemi způsoby chování. Po přesunutí na nová místa se opět zjistí ohodnocení každého jedince a celý cyklus se opakuje, dokud se nenaplní ukončovací podmínka. Před uvíznutím v lokálním extrému je algoritmus vybaven zvyšujícím se vektorem rychlosti, při jehož překonání se vygeneruje nová pozice částice. Aby se částice příliš nevzdalovaly od svého doposud nejlepšího řešení, je částici dovoleno zůstat s již konstantní maximální rychlostí. Další nevýhoda se může projevit na složitých problémech s mnoha optimy, kdy PSO má sklon k předčasné konvergenci. Problémy nastávají i při určitém nastavení všemožných parametrů metody, jakými jsou například setrvačnost, počet částic nebo maximální rychlost [41].

Detailnější přehled o metodách a jejich fungování poskytuje zdroj [38].

### 3 Genetické algoritmy

Evoluční algoritmy jsou stochastické optimalizační metody založené na darwinovském principu evoluce. Pod obecný pojem evoluční algoritmy se obvykle zařazují jako metodologie genetické algoritmy, evoluční strategie, evoluční programování a genetické programování.[11]

Genetický algoritmus pracuje s populací jedinců, přičemž každý z těchto jedinců reprezentuje vhodným způsobem zakódované řešení daného problému. Každému jedinci je přiřazeno nezáporné ohodnocení, které kvantitativně vyjadřuje míru vlastností konkrétního individua vzhledem k řešenému problému, a současně je i měřítkem reprodukční schopnosti. Mluví se tedy o ohodnocení, míře kvality, vhodnosti, síle, reprodukční schopnosti či fitness, kterou chceme k dosažení optimálního řešení maximalizovat.

Genetický algoritmus sám o sobě je slepým algoritmem, neboť kromě ohodnocení jedinců v populaci nevyužívá žádnou apriorní informaci o daném problému či prohledávaném okolí. Tím se tedy stává velmi obecným a univerzálním řešením. Specializované algoritmy pro danou situaci tedy nepředčí ani v rychlosti, ani v kvalitě, avšak dostává se k řešení v situacích, kdy žádná technika není k dispozici, nebo není známa. Je vhodné tedy zdůraznit, že genetické algoritmy jsou svojí podstatou heuristické (cílem je najít přijatelné řešení v přijatelném čase), stochastické (obsahují celou řadu náhodných komponent) a vnitřně lineární (pracují současně s celou populací potenciálních řešení).

Je třeba upozornit, že v naprosté většině prací o genetických algoritmech se pod pojmem chromozom rozumí celý genotyp jedince a ne jen část genetické informace. Chromozom se skládá z lineárně uspořádaných genů, které nabývají různých hodnot. Chromosomem můžeme chápat i symbolickou reprezentaci hodnot parametrů. Hodnoty se nazývají alely a při nejjednodušším kódování nabývají binárních hodnot, tedy 0 nebo 1.

#### 3.1 Teoretický základ genetických algoritmů

Fungování genetických algoritmů lze zjistit z rostoucího fitness jak u nejlepšího jedince, tak u celé nové populace.

Pod pojmem schéma je označen řetězec délky  $l$  nad množinou  $\{0,1,*\}$ , kde znak  $*$  je zástupným znakem. Schéma je jakousi šablonou, ve které místa označená znaky 0 nebo 1 jsou určená a pozice obsahující zástupné znaky se nazývají neurčené. Instance daného schématu je pak řetězec o stejné délce jako schéma a se stejnými znaky na určených pozicích ve schématu. Například ke schématu  $0**1*0*$  můžeme určit instanci ve tvaru  $0001000$ ,  $0111001$  a mnohé další. Počet instancí daného schématu je tedy  $2^k$ , kde  $k$  označuje počet volných pozic.

Řád schématu  $o(H)$  je definován jako počet určených pozic schématu  $H$ , tedy pozic, na kterých se nenachází symbol  $*$ .

Definiční délka schématu  $d(H)$  se chápe jako vzdálenost mezi první a poslední určenou pozicí schématu. Pokud tedy bude schéma  $H = *01**0**$ , pak  $o(H)=3$  a  $d(H)=6-2=4$ .

Schémata a jejich charakteristiky jsou důležitým nástrojem při analýze chování genetického algoritmu, neboť genetický algoritmus pracuje na základě těchto pojmů.

Nechť  $S$  je schéma s alespoň jednou instancí v populaci  $P(t)$ ,  $m(S,t)$  je počet instancí schématu  $S$  v generaci  $t$  a  $u(S,t)$  je střední ohodnocení schématu  $S$  v generaci  $t$  dané vztahem

$$u(S,t) = \frac{1}{m(S,t)} \cdot \sum_{x \in S} f(x) \quad (8)$$

Dostáváme tedy průměrné ohodnocení instancí schématu  $S$  v populaci  $P(t)$ . Cílem je však odhadnout očekávaný počet výskytů instancí schématu  $S$  v generaci  $t+1$ , což bývá označováno jako  $E(m(S,t+1))$ .

Pokud  $f(x)$  bude ohodnocení řetězce  $x$  a  $f(t)$  průměrné ohodnocení populace  $P(t)$ , potom očekávaný počet potomků řetězce  $x$  bude roven podílu  $f(x)/f(t)$ .

$$E(m(S, t + 1)) = \sum_{x \in S} \frac{f(x)}{f(t)} = \frac{u(S,t)}{f(t)} m(S,t) \quad (9)$$

Genetické operátory, které se v genetických algoritmech vyskytují, mohou svými účinky poškodit konkrétní instance schématu  $S$ , a tím snížit počet těchto instancí v populaci. Daná situace se může stát, pokud u jednobodového křížení s pravděpodobností křížení  $p_c$  dojde k tomu, že ani jeden z potomků vytvořených křížením nebude instancí schématu  $S$ . Přitom instance schématu  $S$ , byla jedním z rodičovských chromozomů, na které byl operátor křížení aplikován. Pravděpodobnost destrukce schématu  $p_d(S)$  závisí na definiční délce schématu, neboť ve schématu existuje  $l-1$  míst, kde mohlo ke křížení dojít, a z toho na  $d(S)$  místech mohlo dojít ke zničení schématu.

$$p_d(S) \leq p_c \frac{d(S)}{l-1} \quad (10)$$

Nerovnost vyjadřuje, že při křížení nemusí nutně dojít ke ztrátě instance populaci. Čí je definiční délka schématu větší, tím je menší pravděpodobnost zachování daného schématu i po působení operátoru jednobodového křížení.

Pravděpodobnost mutace  $p_m$  je pravděpodobnost, že dojde ke změně hodnoty na dané pozici. Logicky tedy lze říct, že  $1-p_m$  je pravděpodobnost, kdy ke změně nedojde. Navíc z hlediska zachování schématu jsou podstatné v mutaci jen určené pozice, kterých je  $o(S)$ . Pravděpodobnost, že po mutaci aplikované n instanci schématu S bude i nový řetězec instancí daného schématu, je určena vztahem:

$$p_{s_m}(S) = (1 - p_m)^{o(S)} \quad (11)$$

Vliv mutace na možnou ztrátu instance schématu S je tím menší, čím menší je řád tohoto schématu.

Spojením vlivu selekce a obou genetických operátorů do jednoho vzorce pak dává za vznik odhadu pro očekávaný počet výskytů instancí schématu S v další generaci:

$$E(m(S, t+1)) \geq \frac{u(S,t)}{f(t)} m(S,t) \left(1 - p_c \frac{d(S)}{l-1}\right) [(1 - p_m)^{o(S)}] \quad (12)$$

Obsah tohoto vzorce je zformulovaný do věty o schématech:

„Nechť pro populaci  $P(t)$  platí, že  $m(S,t)$  je počet instancí schématu S v populaci v čase t,  $u(S,t)$  je střední hodnota ohodnocení schématu S a  $f(t)$  je průměrné ohodnocení populace v čase t. Potom očekávaný počet instancí schématu S v populaci  $P(t+1)$  je zdola ohraničený nerovností (12), kde  $p_c$  je pravděpodobnost křížení a  $p_m$  je pravděpodobnost mutace.“

Pokud tedy platí  $u(S,t) > f(t)$  (či  $u(S,t) < f(t)$ ), potom se schéma S vyskytuje v následující generaci s rostoucí (či klesající) pravděpodobností. Odolnější proti destrukci operací křížení budou schémata s menší délkou  $d(S)$  a mutaci budou lépe odolávat schémata s menší hodnotou  $o(S)$ . Tudiž krátká schémata nízkého řádu, jejichž průměrné ohodnocení je větší než průměrné ohodnocení celé populace, mají exponenciálně rostoucí počet instancí v následující generaci. Tím se stále zlepšuje nalezené řešení, aniž by se musely prozkoumávat všechna řešení.

Nejediným rozparem mezi genetickou informací v přírodě a v genetickém programování je skutečnost nadřazenosti alel jednoho genu nad alelami genu jiného. I když se v genetickém programování tento fakt opomíjí, algoritmy správně fungují. Lze to vysvětlit

tím, že genetické algoritmy jsou pouhým přirovnáním, ve kterém si podstatu genů určuje uživatel.

### **3.2 Podmínky pro použití GA**

Genetické algoritmy jsou univerzální, a pokud nejsou přizpůsobeny konkrétnímu problému tak nemusí fungovat ideálně – jiný algoritmus může být mnohem přesnější či rychlejší. Intuitivní indicie pro použití GA jsou následující [11]:

- prohledávaný prostor je velký,
- prohledávaný prostor není vyhlazený a unimodální,
- struktura prohledávaného prostoru je komplikovaná,
- k řešení problému není třeba nutně nalézt globální optimum, ale postačí přijatelné řešení v přijatelném čase.

Pokud prohledávaný prostor bude malý, doporučuje se prohledání celého prostoru k nalezení globálního optima. U vyhlazeného prostoru budou lépe fungovat gradientní metody a při vyhledání optima v prostoru s jednoduchou strukturou se využijí algoritmy, které velmi dobře využijí znalostí o tomto prostoru.

### **3.3 Kódování**

Nejjednodušším typem kódování je použití pouze dvou binárních hodnot. Nevýhoda *binárního kódování* spočívá v nejasnosti změn. Malá změna v chromozomu by se měla projevit jen málo, což ale u binárního kódování neplatí, takže se projeví ve velké míře. Tím se naznačuje, že dva sousední chromozomy mohou být velice odlišné. [11, str42]. Daný problém lze řešit pomocí Grayova kódu, jehož vlastností je zakódování sousedních hodnot tak, aby se lišily pouze v jednom bitu. Lze tedy předpokládat podobné vlastnosti sousedních chromozomů.[19]

Při *permutačním kódování* je jedinec reprezentován pomocí permutace čísel. Permutací je určeno pořadí jednotlivých objektů v řešení daného problému a cílem je vyhledání permutace s optimálním řešením. Známý příklad s permutačním kódováním je u problému obchodního cestujícího [11;19].

Použití kódování pomocí *reálných čísel* lze hledat u algoritmů, kde se vyžaduje velká přesnost. Binární vyjádření zde vede k příliš dlouhým řetězcům, čímž je zapotřebí více paměti a více času na zpracování dat.[19]

Volba způsobu kódování záleží na dané úloze a způsobu jejího pojetí. Nalezení nejlepšího kódování pro danou registraci jak z hlediska funkčnosti, tak z hlediska rychlosti, představuje testování zahrnující různé situace. [2]

**Tabulka 2 Převod binárního kódu na Grayův kód**

Binární kód	Grayův kód
000	000
001	001
010	011
011	010
100	110
101	111
110	101
111	100

### **3.4 Ohodnocení individuí**

Jak již bylo uvedeno, optimalizace se provádí pro vyhledání nejlepšího řešení. V průběhu algoritmu se tedy generují různá řešení problému, z kterých je potřeba správně vybrat to nejlepší. K tomu účelu se využívají ohodnocující funkce neboli fitness, které vyjadřují míru optimalizace. Ohodnocující funkce však musí být v souladu s daným případem, jinak povede k zavádějícímu řešení.

V některých případech binárně kódovaných chromozomů se lze setkat s velmi jednoduchým výpočtem fitness každého jedince. Situace nastane v případě, kdy jsou preferována a považována za zdatnější ta individua, která obsahují ve svém chromozomu více jedniček. Výpočtem fitness hodnoty může být součet alelových hodnot každého jedince, což je stejný výsledek jako počet jedniček v daném chromozomu.

Nejčastěji se lze setkat se vzorci, které určí fitness hodnotu. Tři případy těchto funkcí uvedla kapitola 1.3, ve které se zmínilo vypočítání pomocí sumy rozdílů čtverců, normovaného korelačního koeficientu a vzájemné informace.

V případech, kdy míra ohodnocení je pouze relativním měřítkem, musí se využít jiných přístupů k výpočtu fitness, než jaké zde již byly uvedeny. Příkladem s relativním měřítkem se lze chápat případ, ve kterém dvojice jednotlivců spolu soupeří a relativní vítěz každého souboje získá bod. Ztrácí se tudíž informace o absolutním vítězi, neboť i slabší jedinec by mohl získat spoustu bodů díky souboji se slabšími jedinci. Dojít ke správnému řešení umožní turnaj mezi individui. Vytvoří se soupeřící dvojice individuí a lepší jedinec postupuje do dalšího kola, ve kterém se bude opakovat stejný děj. Tím se dospěje k vítězi a ostatním jedincům se přiřadí odpovídající hodnota v turnaji. [11]



Zvláštním případem ohodnocující funkce je *interaktivní evoluce*, kdy sám uživatel může zasahovat do ohodnocení jednotlivých případů. V praxi byla tato metoda vytvořena ve spolupráci Caldwell a Johnstone a využívá se v kriminalistice, kdy je zapotřebí určit viníka trestného činu. Svědek na základě zapamatovaných rysů obličej pachatele zasahuje do ohodnocení, čímž je schopen v krátkém čase určit z 34 miliard různých typů obličej ten nejpodobnější viníkovi. Celý algoritmus je založen na kombinaci pěti základních stavebních znaků - typ a pozice vlasů, očí, tvar a umístění nosu, úst a brady. Provádí se vygenerováním počáteční populace o velikosti dvaceti obličejů a ohodnocením daných charakteristik. Tím se provede selekce pro jednotlivé znaky a po aplikaci genetických operátorů mutace a křížení se opět vytvoří další sada obličejů. Je dokázáno, že pouhých deset generací postačí k určení dostatečné podoby obrazu s viníkovou tváří.

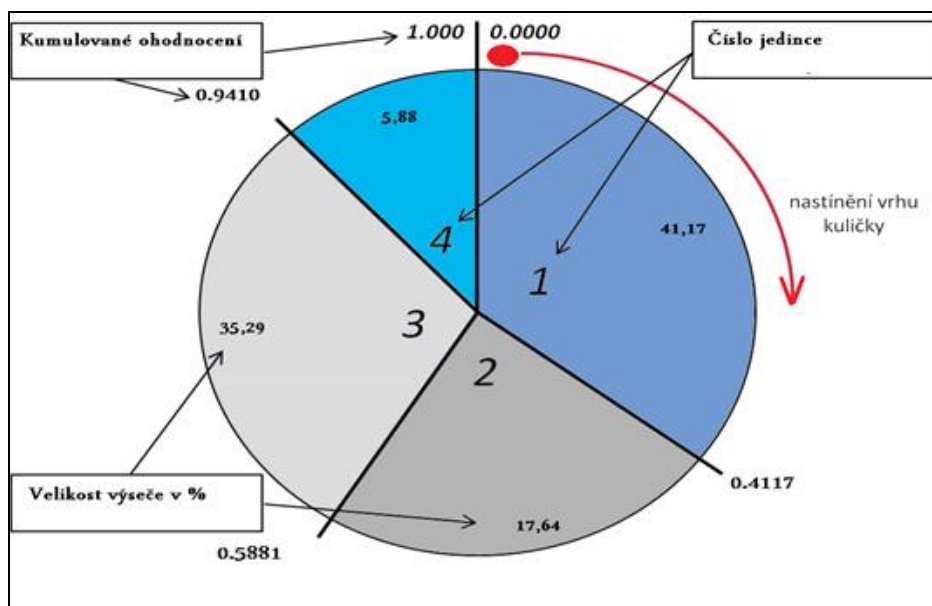
### 3.5 Selekcce

Pro vytvoření nové populace o stejné velikosti jako má počáteční populace (například 50 jedinců [27]), je zapotřebí náhodného výběru jedinců. Jedinci budou sice vybíráni náhodně, avšak pro konvergenci ke správnému řešení je zapotřebí, aby se kvalita jedinců zvyšovala (zvyšovalo se fitness ohodnocením). Inspirací u genetických algoritmů je samotná příroda, kdy větší nárok na přežití mají zvýhodnění tvorové. Z tohoto důvodu se aplikují různé mechanismy, které zohledňují zdatnost jedince.

*Turnajový výběr* vybere z populace náhodně  $k$ -soutěžících (nejčastěji  $k=2$ ) a z nich vybere nejlepšího. Tento postup se opakuje. Přežívají tedy nejlepší jedinci, avšak dochází i k zachování diverzity, neboť do souboje jsou vybíráni jedinci náhodně. Při větším hodnotě parametru  $k$ , se bude uplatňovat větší selektivní tlak, čímž k další reprodukci nebude připuštěno více jedinců [11].

Naproti tomu *inverzní turnajový výběr* zvolí z populace náhodně  $k$ -soutěžících (nejčastěji  $k=2$ ) a z nich zahodí nejhoršího. Tento postup se opět opakuje  $n$ -krát, kde  $n > p$  přičemž  $p$  je počet výběrů řešení. V tomto mechanismu selekce je zaručeně vybrán nejlepšího jedince. Ovšem existuje i pravděpodobnost vybrání nejhoršího řešení tím, že se nedostane do turnaje.

*Ruletový mechanismus* je založen na přirozeném výběru, kdy větší pravděpodobnost pro přežití je dána zdatnějším jedinci s vyšší hodnotou fitness. Jelikož ale jde o náhodný proces, mají možnost uspět i nejslabší jedinci. Pro vytvoření ruletového kola postačí sečíst všechna ohodnocení a proporcionálně každému jedinci přiřadit kruhovou výseč v závislosti na velikosti jeho fitness, jak je to znázorněno na Obrázku 3. Ruletový mechanismus a turnajový výběr patří mezi nejčastěji používané selekce [14].



**Obrázek 3 Ruletový mechanismus [44]**

*Stochastické univerzální vzorkování* představuje modifikaci ruletového mechanismu. Avšak namísto několika výběrů se provede jen jeden výběr několika jedinců stejně vzdálených v ruletovém kole. Pro představu u ruletového mechanismu jedním roztočením kola kulička vybere jednoho jedince. Zde však je vysláno  $k$ -kuliček, které mají od sebe konstantní vzdálenost, a roztočení kola proběhne jen jednou. Dostane se  $k$ -jedinců podle toho, kde se kuličky po jednom roztočení kola zastavily. Mechanismus tedy zaručuje rozmanité celočíselné očekávané hodnoty. Z důvodu neopakování roztočení rulety se pokládá za výpočetně nejméně náročnou metodu. Avšak pro příliš velké ohodnocení pouze několika jedinců neřeší problém předčasné konvergence k lokálnímu optimu.[11]

*Pořadová selekce* se používá v případě, kdy rozdíly hodnot jedinců jsou příliš výrazné. To vede k příliš častému zastoupení úspěšnějších jedinců a rychlé konvergenci k suboptimálnímu řešení. Při tvorbě se sestaví žebříček podle ohodnocení, takže nejlepší jedinci sice jsou více zastoupení, ale přitom jen úměrně svému pořadí, a vytvoří se ruletový mechanismus s novými pravděpodobnostmi zastoupení. Mechanismus bude lépe fungovat i v případě, kdy budou převládat velmi silní jedinci s nepatrnými rozdíly jejich ohodnocení, avšak na úkor času. Selhávat bude mechanismus při opačném případě použití, tedy pokud budou k dispozici velmi podobní jedinci, neboť vytvoří v jejich ohodnocení velké rozdíly. Tento efekt lze zmírnit použitím dalších vztahů [11, 35].

Při vhodně zvolených parametrech lze docílit všemi mechanismy podobných výsledků, jak udává práce [9], která testovala poměrnou reprodukci, turnajový výběr, pořadovou selekci a selekci stálého stavu. Poměrná selekce v sobě zahrnuje například ruletový mechanismus nebo stochastické univerzální vzorkování. Selekcí stálého stavu je založená na pořadové selekci, kdy se nahradí pouze několik individuů.

K různým mechanismům lze přidat *elitismus*, což je rozhodnutí pro postoupení jednoho či více nejlepších jedinců do další populace, aniž by museli projít nějakým dalším mechanismem selekce. Tím se urychlí konvergence. [18]

### 3.6 Genetické operátory

K vytvoření potomků dvou jedinců mohou být využívány dva základní genetické operátory: křížení a mutace.

#### Křížení

Při průběhu operátoru křížení vstupují do procesu vždy dva jedinci, kteří si vymění část svého chromozomu, a tím dají vzniknout novým jedincům s novou kombinací alel. Každý potomek má tedy určitou část od prvního rodiče a zbylou část od druhého rodiče. Tento proces nastává obvykle s 0,75-0,95% pravděpodobností [11], takže existuje malá pravděpodobnost, že by křížení nenastalo a do další generace by se přenesly individua beze změny. Literatura [27] používá koeficient křížení roven 0,82. V optimalizaci se efekt křížení může projevit odpoutání se od oblasti s lokálním minimem.

*Jednobodové křížení* spočívá v náhodném zvolení jednoho genu v chromozomu a následné výměně zbylé části chromozomů mezi dvěma jedinci. Při *k-bodovém křížení*, kde  $k > 1$  se náhodně vygeneruje  $k$  míst ke křížení, čím vznikne  $k+1$  částí chromosomů. V průběhu působení operátoru se vždy postupuje zleva a při dospění k místu křížení si dva chromosomy vymění všechny geny napravo. Postup se opakuje dokud se nedospěje na konec chromosomů. *Uniformní křížení* je zobecněním *k-bodového křížení*, kdy se pro každý gen v chromosomu s 50% pravděpodobností rozhodne, zda bude genem 1. nebo 2. potomka.

U reálných čísel lze vytvořit i další operátory křížení. *Aritmetický operátor křížení* bude novou alelu určovat aritmetickým průměrem rodičovských alel, *geometrickým operátorem* se nová alela vytvoří odmocněním součinu rodičovských alel atd.

Křížení permutačně zakódovaných čísel provází následná úprava, neboť po křížení nejsou permutacemi. Používanou technikou je PMX (operátor *křížení s částečným přiřazením*)[44]. Nejprve se vymění podřetězce určené pro křížení a alely ostatních genů se zanechají neurčené. Na tyto neobsazené pozice se přepíše pouze ty hodnoty genů po rodičích, které nezpůsobí kolizi permutačního kódování. Na zbylé místa se aplikují přepisovací pravidla, takže nově vzniklý jedinec je opět permutačně kódován.

#### Mutace

Mutace působí na chromosomy s relativně malou pravděpodobností většinou v rozpětí 0,001-0,05% [11] a mění náhodně hodnotu jednotlivých genů. Tím brání k příliš rychlému zjednotvárnění vlastností populace, ztrátě potenciálně užitečné genetické informace a také

k předčasné konvergenci populace. Četnost mutace lze adaptivně upravovat v závislosti na výsledcích průběhu registrace. Tím se rozumí zvyšování pravděpodobnosti mutace při stagnující populaci a při lepších výsledcích navrácení pravděpodobnosti na nižší hodnotu. Do proměnné mutace lze zařadit i postup, kdy se zvolí velká hodnota mutačního operátoru při inicializaci a postupně se bude snižovat. Tím se docílí lepšího prohledávání celého prostoru na počátku a při nízké pravděpodobnosti se mechanismus detailněji zaměří na oblasti s nejlepšími výsledky.[44]

Při *binární mutaci* dochází k výměně jedné alely za druhou. Pokud tedy proběhne mutace, lze se stoprocentní pravděpodobností určit hodnotu alely.

Operátor mutace lze různě modifikovat k použití pro reálná nebo celá čísla. Při *náhodné výměně* se původní alela nahradí náhodně vygenerovanou hodnotou, při *aditivní změně* se k původní alele přičte náhodně vygenerovaná hodnota, nebo při použití *multiplikativní změny* se náhodně vygenerované číslo násobí s původní hodnotou alely.

Při aplikaci operátoru mutace na individua s permutačním kódováním musí být provedena mutace s ohledem na zachování permutace. Při *výměnné mutaci* se vyberou náhodně dvě místa, u kterých se vymění hodnoty genů. Tím se řešení vyhne opravným mechanismům, použitým při vynechání či zmnožení alely. Podobná situace je při změně pozice pouze jednoho vybraného genu na zvolené místo v chromozomu. Mutace se nazývá *posuvná*. Dalším operátorem využívaným zvláště u permutačního kódování je *inverzní mutace*, který pouze invertuje úsek genů mezi dvěma vybranými geny. Jelikož se pouze změnilo pořadí, opět není nutné použití žádných opravných mechanismů. Inverze se však často neuvádí, protože podle některých studií bylo prokázáno, že pouze prodlužuje čas výpočtu, aniž by zvyšovala jeho účinnost. [11]

Zajímavou mutací je tzv. *selektivní mutace*. Pokud k takové mutaci dochází, zkoumá se kvalita příslušného jedince před mutací a po mutaci a mutace se povolí pouze tehdy, pokud danou změnou dojde ke zlepšení kvality.

### **3.7 Ukončující podmínka**

Probíhání cyklů s vytvářením potomků probíhá tak dlouho, dokud se nedosáhne ukončovací podmínky. Touto podmínkou může být dosažení určité hodnoty, která se nastavila při inicializaci parametrů, maximální zadaný počet proběhlých generací [14] či nalezení uspokojivého řešení.

### **3.8 Paralelní genetické algoritmy (PGA)**

Účelem paralelizace genetického algoritmu je urychlení procesů, neboť populace mají velké množství jedinců a samotný průběh hodnotící funkce bývá z časového hlediska

náročný. Nově vytvořený Paralel Computing Toolbox Matlabu nabízí účinné nástroje pro dosažení hodnotných výsledků paralelizace. [22]

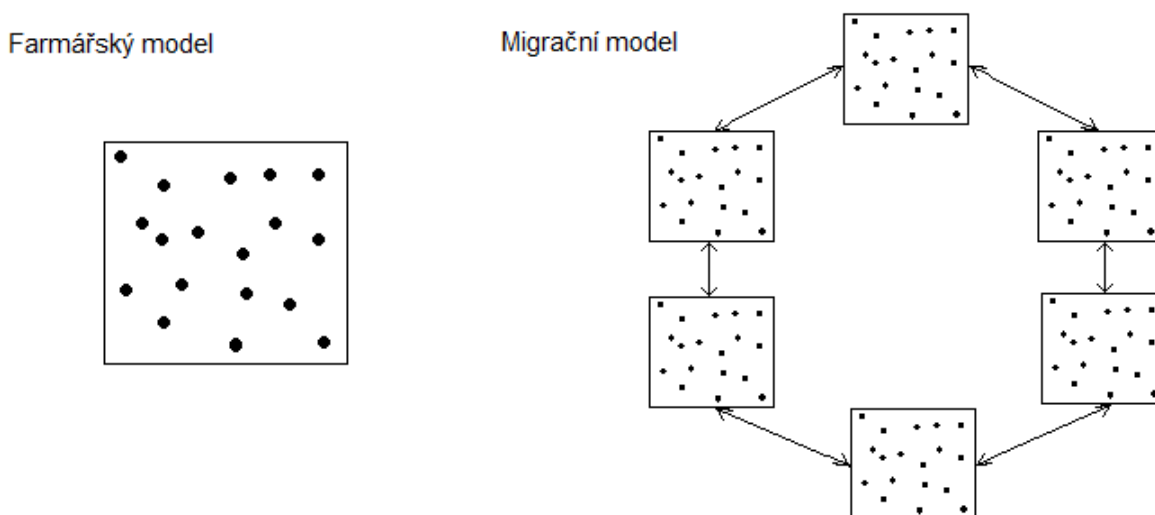
### **Farmářský model**

Při paralelizaci na úrovni výpočtu ohodnocující funkce se na podřízené procesory přivedou jedinci a provede se ohodnocení, jehož výsledky se vrátí na řídicí procesor. Tento přístup se nazývá *paralelizace globálního modelu, farmářského modelu* neboli „*master-slave*“, kdy pánem se rozumí hlavní proces řídicí všechny podřízené procesy označované jako otroky. Protože se uvažuje jen jedna populace v modelu, selekce bere v úvahu všechny jedince, takže každý chromosom má šanci reprodukovat se všemi ostatními. Ohodnocení či použití genetických operátorů na různé jedince však může trvat rozdílnou dobu, proto je důležité snažit se o rovnoměrné rozdělení výpočetní zátěže na jednotlivé podřízené procesory. V případě synchronního algoritmu se v každé populaci čeká na ohodnocení všech jedinců na podřízených procesorech a teprve po přivedení všech ohodnocení na hlavní procesor algoritmus na hlavním procesoru pokračuje dále. Asynchronní algoritmus, který nečeká na zpětnou reakci, ale provádí selekci již při vracejících se odpovědích z jednotlivých podprocesorů se příliš často nevyužívá. Farmářský model poskytuje dobré výsledky v případě, kdy doba potřebná pro komunikaci je zanedbatelná vzhledem k době potřebné pro výpočty mutace, křížení a ohodnocení [41].

### **Migrační model**

V druhém případě tzv. *vícekmenném PGA* se na podřízené procesory přivádí podpopulace (kmény). Jelikož se kmény vyvíjí samostatně, může se použít i rozdílných parametrů, rozdílných metod pro selekci či reprodukčních operátorů. Výhodou tohoto tzv. *migračního modelu* či někdy mylně označovaného (není vždy splněna podmínka úplného propojení podpopulací) *ostrovního modelu* je fakt, že kmény konvergují rozdílným způsobem, čímž prozkoumají různé oblasti v prohledávaném prostoru, a tím zabrání předčasné konvergenci. Čas od času si však některé kmény mohou vyměňovat s určitou pravděpodobností své jedince [21]. Je tedy nutné určit komunikační topologii (způsob propojení mezi jednotlivými kmény), frekvenci migrací, typ výběru a počet migrujících jedinců, rozhodnutí, zda kopie migrujících jedinců zůstanou v kmenu, zda bude vstupní podmínka do kmenu a jak se provede nahrazení v novém kmenu. Vliv těchto rozhodnutí je popsán v [11, 29, 41], kde jedním z mnoha doporučení k dosažení dobrých výsledů je nastavení malého počtu migrantů se střední délkou epochy (5-10% z celkového počtu generací). Předpoklad urychlení algoritmu v tomto druhém případě je v některých případech splněn více, než by se dalo výpočetně očekávat, tedy že výpočetní proces se při použití K procesorů urychlí maximálně K-krát. Avšak může se to dít na úkor kvality nalezeného řešení. [29]

Extrémem migračního modelu je *difusní model*, ve kterém se do každého procesoru soustředí pouze jeden jedinec. V každé generaci si každý jedinec vyhledá ze svých sousedů partnera vhodného k reprodukci, s kterým je dále kombinován. [41]



Obrázek 4 Farmářský a migrační model paralelních genetických algoritmů

### 3.9 Hybridní genetické algoritmy

Větší šance GA spočívá v jejich kombinaci s klasickými technikami. Pojem hybridizace označuje proces adaptace GA tak, aby se zvýšila jeho efektivnost pro určitou třídu problémů. Produktem této hybridizace jsou hybridní GA. Hybridizací tedy rozumíme zahrnutí nejlepších částí tradičních algoritmů, navržených pro danou třídu problémů, do algoritmu genetického. Autoři hybridních GA si sami určují podmínky a vytváří algoritmus kreativním způsobem. Například výsledky spojení GA a horolezeckého algoritmu [33] sebou přináší mnohem rychlejší konvergenci ke správnému výsledku.

Autoři práce [12] se zabývali hybridní PSO, která využívá subpopulaci a křížení genetických algoritmů. K registraci využili sady trojrozměrných dat získaných z magnetické rezonance a počítačové tomografie. Pro srovnání rigidní transformace (translace a rotace ve směru tří os) použili v multimodálních medicínských obrazech často využívanou funkci vzájemné informace. Částice jsou rozděleny do několika podpopulací, ve kterých probíhá PSO zvlášť. Každá podpopulace má své vlastní optimum. Pokud je toto podoptimum lepší než celkové optimum, celkové optimum je nahrazeno tímto podoptimumem. Následuje křížení, při kterém se vyberou ze dvou subpopulací dvě nejlepší podoptima a jejich hodnoty se zkříží pomocí aritmetického křížení. Nejhorší jedinci z daných dvou populací jsou následně nahrazeni takto nově vytvořenými potomky. Výsledky experimentů jsou zahrnuty ve dvou tabulkách s rozdílnými hodnotami

transformačních parametrů, ze kterých lze vyčíst hodnoty registrace pro posuny a rotace jednotlivých os. Dokázalo se, že navržená hybridní PSO dává mnohem lepší výsledky než samotné metody jednotlivě.

## 4 Využití globálních optimalizačních metod pro registraci medicínských obrazů

Teorie registrace obrazů se dnes bohatě využívá u různých lékařských přístrojů, jakými je ultrazvuk, rentgen, výpočetní tomografie (CT), pozitronová emisní tomografie (PET) či nukleární magnetická rezonance (NMR).

Užití teorie registrace v medicínské diagnostice se může rozdělit do čtyř skupin, v závislosti na způsobu sběru dat. [42]

### 1) Sběr dat z různých úhlů pohledu

Snímky stejné scény jsou snímány z různých úhlů pohledů. Cílem je získání hloubky obrazu, většího dvojrozměrného obrazu, či trojrozměrné struktury dat. Tento proces tedy směřuje k fúzování, kdy je snaha vytvořit tzv. panoramatický snímek nebo prostorový snímek.

Při sběru dat z různých úhlů pohledu retiny se vytváří snímek sítnice, který zahrnuje velkou část sítnice, až 80%. Takového rozsahu by se normálním snímáním s mydriatickou kamerou nedosáhlo.

### 2) Sběr dat v různém čase

Snímky stejné scény jsou snímány v časových intervalech s určitou prodlevou. To může vést i k rozdílným podmínkám při snímání. Cílem je najít a ohodnotit změny, které proběhly na snímcích po sobě následujících, a tím získat informace o časovém vývoji.

Při sběru dat v různých časových intervalech se snímky ze stejných lékařských zobrazovacích modalit přes sebe překryjí a sledují se diference. Tímto způsobem může být monitorována progresse chorob jako například růst tumoru, nebo účinnost zdravotní terapie.

Videofluoroskopie je speciálním případem skiaskopie se zaměřením na vyšetření polykacího traktu za účelem získání dynamického záznamu polykání. Využívá se dvou značek umístěných v zadní části dutiny ústní a v hltanu. Při průchodu bolusu v místě značek se zaznamenává časový profil a určuje se míra postižení. Výpočetně metoda stojí na genetických algoritmech. [4]

### 3) Sběr dat různými senzory

Lícování se nemusí omezovat pouze na snímky ze stejného lékařského přístroje, ale může lícovat i snímky jedné scény z různých zobrazovacích modalit. Spojením takto získaných dat do jedné scény, se získají komplexnější a detailnější informace,



neboť každá modalita dokáže díky svému mechanismu snímat pouze omezený druh informací.

Při použití snímku z PET a CT získáme informaci o rozložení radionuklidu (pomocí PET) v přesně situované části těla (pomocí CT), takže propojíme funkci se strukturou. Zjistí se tedy informace o pacientovi, růstu tumoru nebo se ověří správnost prováděné léčby.[6]

Práce [31] je zaměřena na registraci trojrozměrných dat z CT a MRI pomocí genetických algoritmů. Fúzí dat z těchto modalit se docílí dobré vizualizace mozku, kdy data z CT informují o struktuře díky dobrému kontrastu kostí, zatímco data z MRI se zaměřují na měkké tkáně, jakými jsou bílá a šedá hmota mozková. Upozorňuje se i na fakt jemné distorze MRI senzorů, kdy tedy nelze použít jednoduché rigidní transformace. Po ukončení prohledávání prostoru se doporučuje následné zpracování, kdy se pro nejlepší řešení vyhledá lokální extrém, který bude zároveň extrémem globálním.

Dalším případem může být vědecký článek z univerzity Chang Gung [10], ve kterém autoři navrhli adaptivní genetický algoritmus. Pro registraci se vyberou data CT obrazu a s nimi korespondující data povrchu obličeje. Obě sady dat se z důvodu velkého objemu, který by se negativně projevil na rychlosti zpracování dat, zredukuje na 300 hodnot. Adaptibilita metody spočívá v použití upravené mutace. Pokud je nejlepší řešení vzdálené může mutace probíhat rozsáhleji. Jakmile se však konvergence zpomalí, mutace zmenší svůj rozsah, aby se i nadále zůstalo poblíž správného řešení a našel se nejlepší chromosom. Výsledky experimentu ukazují, že i když fitness hodnota je v obou případech srovnatelná, adaptivní GA dosahují cíle s mnohem menším počtem generací (v průměru padesáti pokusů až o jednu třetinu) než klasické GA.

Práce [24] se zabývá současnou registrací trojrozměrných dat ze tří modalit: PET, CT a SPECT. Transformační matice je získána užitím kalibračního fantomu a rámu podložky pro laboratorní zvířata, na kterých se experiment provádí. Značení na zvířatech pomocí externích značek se nedoporučuje, neboť je nespolehlivé, těžce kalibrovatelné a může negativně interferovat se sadou dat. Tímto způsobem prováděné in vivo studie umožňují rozvoj farmaceutik, testování nových terapeutických přístupů léčby lidských nemocí či zobrazení genové exprese.

Nejčastěji zmiňované registrace MRI-CT s sebou přináší velkou přesnost, avšak mezi jejich nevýhody patří vysoká cena operačního systému, ztížené použití při operacích a nadbytečná radiace při použití CT. Proto se vymyslela registrace využívající předoperačních obrazů CT a dat získaných ultrazvukovou sondou při operaci, u které se zmíněné nevýhody nevyskytují. Uživatel je však kvůli šumu

omezen jen na takovou kvalitu snímků, při které jsou dobře rozeznatelné jen kostní struktury. Proto se tohoto nástroje může využívat jen v omezených případech, jako například operace páteře.[39]

Byla vytvořena i registrace trojdimenzionálních dat získaných pomocí MRI představující model srdce a sady histologických řezů. Tím se zajistilo výborného rozlišení srdečních struktur v srdci myši. [23]

#### 4) Registrace scény v modelu

Předpokládá se, že je k dispozici snímek scény i model scény. Modelem může být počítačová reprezentace dané scény, jako například mapa, nárys nebo průměrný vzorek. Cílem je nalezení snímku scény v modelu a jejich následné srovnání. [40, 28]

Užitím lícování snímku z určitého přístroje do modelu se můžou porovnávat patientské snímky s digitálním anatomickým atlasem [13, 15]. Toho se využívá i při klasifikaci objektů. V práci [17] se v medicínských snímcích určují jak kvantitativně, tak kvalitativně krevní buňky. Před samotným genetickým algoritmem se snímky předzpracovaly segmentací metodou prahování, přičemž daných práh se určil z histogramu obrazu. Účinnost popsané metody s danými parametry je 94-100%.

Registrace nemusí být využita jen v „pasivním“ získávání a srovnávání dat, ale také může být využita v procesu, ve kterém se okamžitá hodnota využívá pro rozhodování dalšího postupu. Příkladem mohou být přesné robotické lékařské vrtačky, které pracují na základě průběžně získaných snímků z CT.

## 5 Navržení optimalizační metody

Při navrhování optimalizační metody pomocí genetického algoritmu se využijí klasické postupy. Aplikovat se budou na datové soubory, které představují hodnoty intenzit dvou obrazů. V průběhu zkoumání kvality navrženého algoritmu bude docházet ke změně hodnot vstupních parametrů jakými jsou pravděpodobnost mutace, pravděpodobnost křížení, velikost dat či maximální počet generací.

Získání dvou datových souborů reprezentující hodnoty intenzit dvou obrazů.

1. Inicializace parametrů.
2. Vynulování hodnoty počítadla generací  $t=0$ .
3. Vytvoření počáteční populace jedinců  $P(t)$  reprezentovanými transformacemi v chromozomu.
4. Zjištění společných bodů těchto obrazů každého individua.
5. Vypočítání ohodnocující funkce (fitness) každého individua.
6. Selektce jedinců pro vytvoření nových jedinců.
7. Aplikace operátorů křížení a mutace na vybrané jedince.
8. Vytvoření nové populace  $P(t+1)$  z rodičů a jejich potomků.
9. Zjištění společných bodů těchto obrazů každého individua.
10. Vypočítání ohodnocující funkce každého individua.
11. Pokud je splněna ukončovací podmínka, výsledkem ohodnocení populace ( $t$ ), jinak se pokračuje bodem číslo 5.

## 6 Testovací data

Kvalita vytvořeného optimalizačního algoritmu se hodnotí pomocí srovnávání výsledků různých algoritmů nebo sledováním dosaženého extrému na předem známých testovacích funkcích. Třetí možností je vytvoření testovacích obrazů, jejichž líčováním získáme kritériální funkci, kterou opět podrobíme vytvořenému algoritmu a sledujeme kvalitu optimalizace. Jelikož srovnávání výsledků algoritmů je podstatou navazující části této práce, tato kapitola se budeme věnovat testovacím funkcím a testovacím obrazům.

### 6.1 Testovací funkce

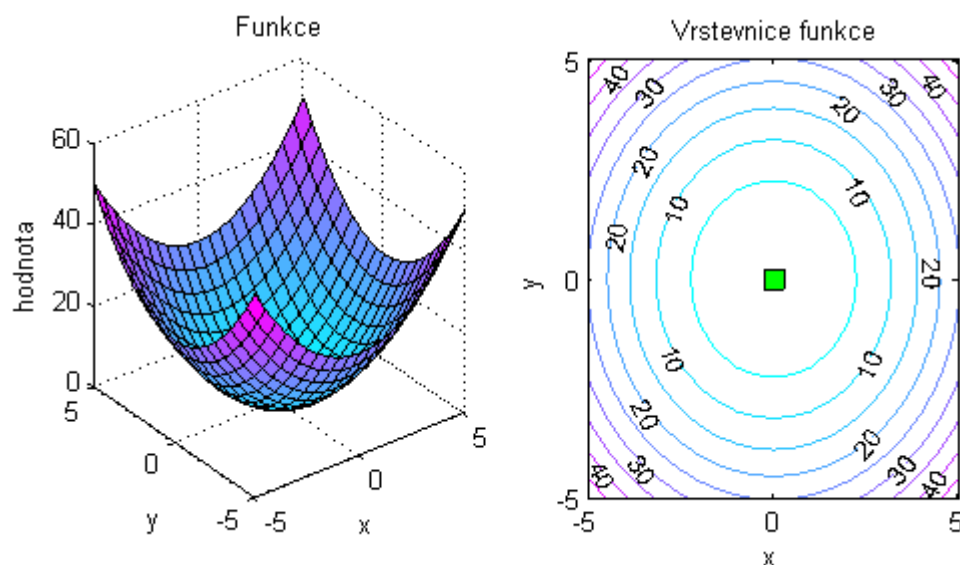
Testovací funkce obsahují širokou škálu vlastností, jakými jsou nelinearita, spousta lokálních extrémů, rovina okolo jednoho úzkého extrému apod. Optimální hodnota všech funkcí zmíněných v této podkapitole se bude nacházet v globálním minimu. Funkce budou znázorněny v takovém rozsahu hodnot  $x$  a  $y$ , aby byly co nejlépe vidět jejich vlastnosti. Při samotném testování však bude stanoven pevný rozsah pro všechny funkce k získání hodnotného porovnání výsledků. Budou uvedeny i vrstevnicové grafy s vyznačeným globálním minimem. Vykreslení funkcí získáme pomocí Matlabu R2007b.

#### První De Jongova funkce

Funkce je jednou z nejjednodušších testovacích funkcí. Díky své konvexnosti a jednomu extrému je vhodná pro velké množství optimalizačních technik. Její předpis se definuje [26]

$$f(x) = \sum_{i=1}^n x_i^2 \quad (13)$$

kde  $n$  označuje počet proměnných, tedy v tomto případě počet souřadných os.



**Obrázek 5 První De Jongova funkce (vlevo) s vrstevnicovým grafem (vpravo)**

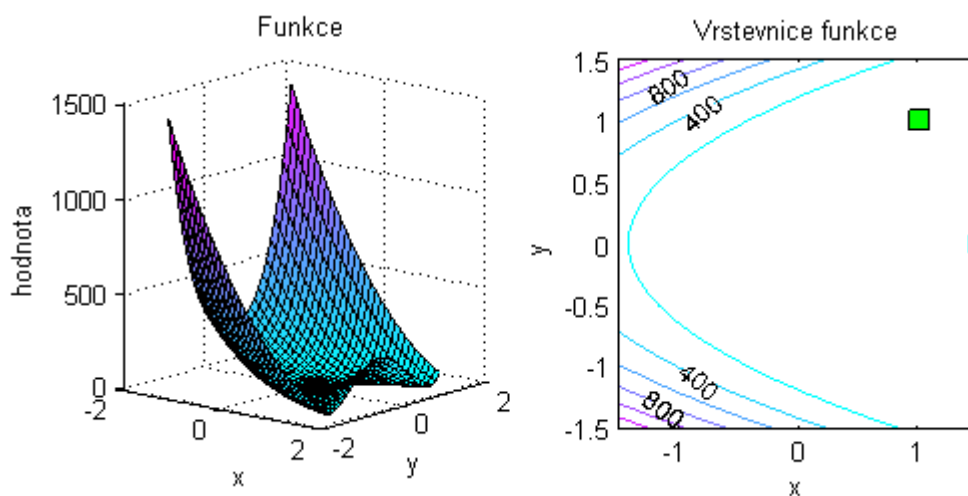
Na Obrázku 5 vlevo je naznačen průběh funkce v rozsahu  $\pm 5$  a krokem 0,5. Pro větší přesnost je potřeba zvolit dostatečně malý krok a větší rozsah, avšak za cenu časové náročnosti a většího množství dat. Průběh funkce se v tomto případě při změně rozsahu nezmění, neboť se jedná o paraboloid. Globální minimum se nachází v počátku soustavy souřadnic, což označuje zelená značka v Obrázku 5 vpravo.

### Resenbrockovo sedlo

Známa i pod názvem banánové údolí má Druhá De Jongova funkce zvláštní tvar, který můžeme vidět na Obrázku 6. Globální maximum funkce leží v dlouhém úzkém údolí na pozici  $(x,y) = (1,1)$ . Předpis funkce je [26]

$$f(x) = \sum_{i=1}^{n-1} [100(x_{i-1} - x_i^2)^2 + (1 - x_i)^2]. \quad (14)$$

Pro větší rozsah hodnot  $x,y$  vznikne údolí s téměř konstantní hodnotou, neboť krajní hodnoty funkce osy  $y$  velmi rychle konvergují k nekonečnu. Daný jev zapříčiní obtížné nalezení přesné hodnoty globálního optima pro většinu optimalizačních funkcí.

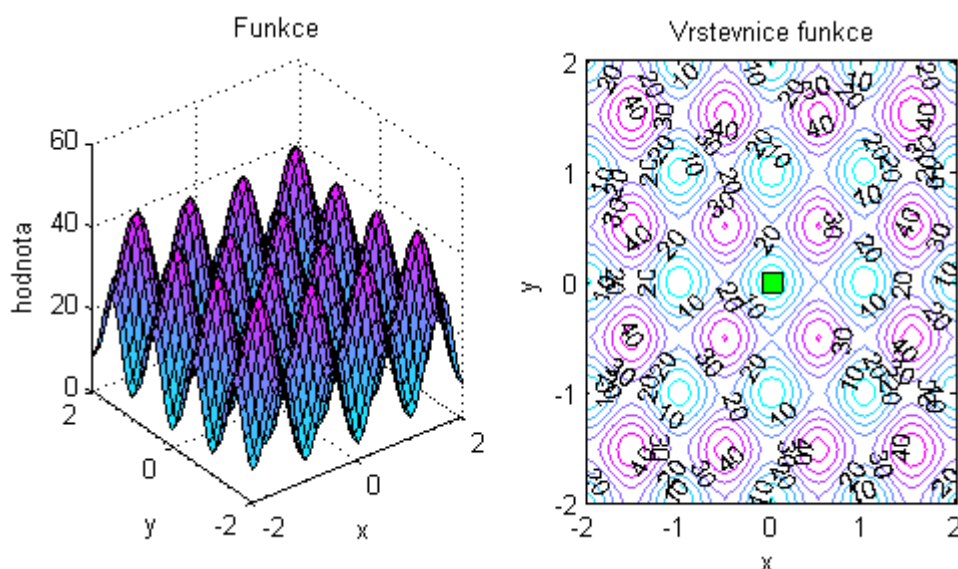


Obrázek 6 Resenbrockovo sedlo (vlevo) s vrstevnicovým grafem (vpravo)

### Rastriginova funkce

Rastriginova funkce je založena na První De Jongově funkci, ke které se přidá kosinový průběh. Vznikne tedy velké množství lokálních minim, které jsou však zřejmé jen při malém rozsahu a jemnějším kroku jak je to ukázáno Obrázkem 7. Při větším rozsahu (například  $\pm 50$ ) se funkce bude jevit ryze jako První De Jongova funkce. Globální minimum se tedy opět nachází v počátku soustavy souřadnic. Nalezení globálního minima je kvůli velkému množství lokálních extrémů náročné. Funkce je definována [26]

$$f(x) = 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)] \quad (15)$$

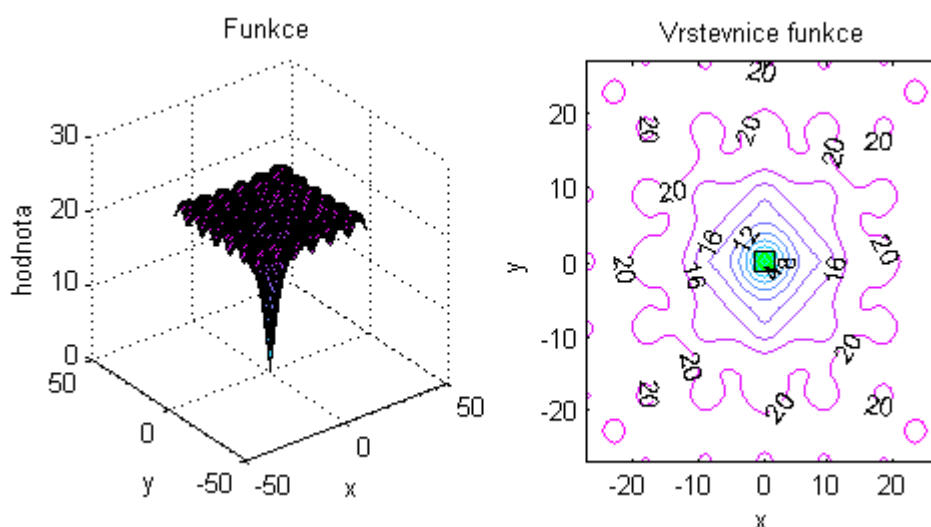


Obrázek 7 Rastriginova funkce (vlevo) s vrstevnicovým grafem (vpravo)

## Ackleyho funkce

Ackleyho funkce je často užívanou funkcí se spoustou lokálních extrémů. Oproti předchozí Rastriginově funkci však mezi hodnotami lokálních minim v počátku souřadného systému je větší rozdíl a ve velkých vzdálenostech od středu se funkce jeví jako rovina. Obrázek 8 ukazuje hrot s globálním minimem a zároveň obsahuje i vzdálenější hodnoty, které naznačují začátek nekonečné roviny. Definice funkce je [41], kde  $e$  znamená konstantu o velikosti 2,71828.

$$f(x) = 20 + e - \sum_{i=1}^n \left[ \frac{20}{e^{0,2 \sqrt{\frac{(x_i^2)}{2}}}} - e^{(0,5(\cos(2\pi x_i)))} \right] \quad (16)$$



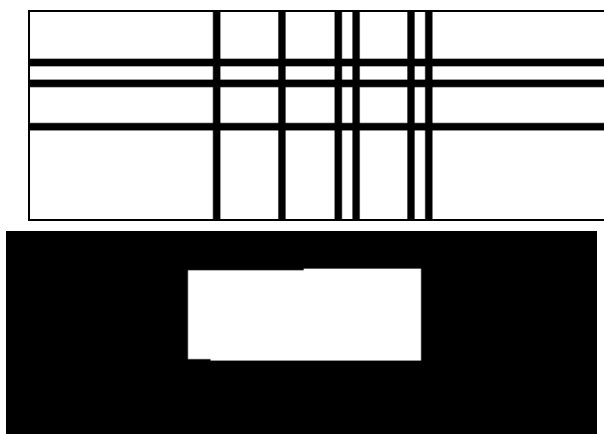
Obrázek 8 Ackleyho funkce (vlevo) s vrstevnicovým grafem (vpravo)

## 6.2 Testovací obrazce

Podobně jako u testovacích funkcí, testovacími obrazci lze hodnotit kvalitu algoritmu. Musí se však znát správná hodnota globálních extrémů. Na rozdíl od již zmíněných testovacích funkcí, identické testovací obrazce představují jak registrační, tak testovaný obraz. Podle výběru kritériální funkce registrace se optimum nachází v minimu či maximu funkce.

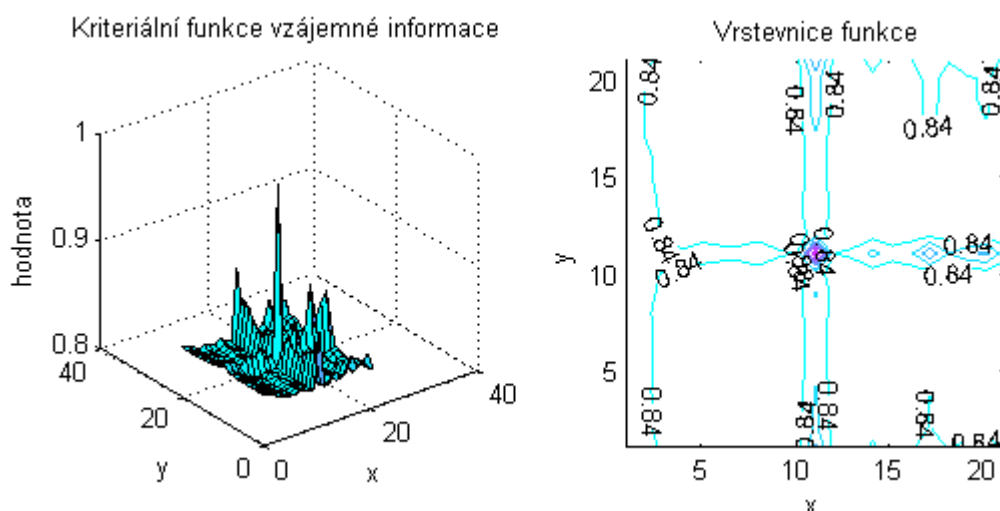
Vytvořili jsme si dva testovací obrazce. Jeden je složený z osově rovnoběžných pruhů s nepravidelnou vzdáleností (Obrázek 9 vlevo). Na druhém obrazci jsou rozmazané tečky s kartézským rozložením (Obrázek 11 vlevo). V programovém prostředí Matlab jsme si vytvořili algoritmus pro lícování obrazů. K dosažení ideálního slícování se použili dva identické obrazce. Jelikož lícování probíhalo transformacemi jednoho obrazce po druhém, oblast zájmu posunovaného obrazce se označila pomocí masky (Obrázek 9 vpravo,

Obrázek 11 vpravo). Pro vytvoření dvourozměrné funkce se využila transformace v ose x a ose y v rozsahu  $\pm 50$  pixelů od polohy, ve které jsou obrazy ideálně slícované. U obou testovacích obrázců se tedy dosáhlo nejlepšího slícování ve středu dvourozměrné funkce. Pro přehlednější obrázky se k výpočtu používal každý pátý pixel. Kriteriaální funkcí byla zvolena vzájemná informace, neboť dosahovala názornějšího dosažení extrému než funkce korelačního koeficientu. Názornějším se rozumí obsah užších špiček v bodech, ve kterých se transformacemi dosáhlo dobrého lícování.



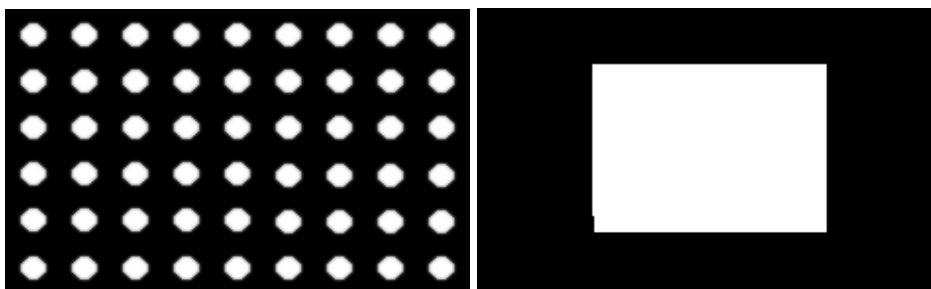
**Obrázek 9 Testovací obrazec složený z pruhů (vlevo) a jeho maska (vpravo)**

Vrstevnice funkce na Obrázku 10 (vpravo) naznačují vyšší hodnoty lícování v horizontálním a vertikálním směru, což odpovídá pruhům v identickém směru.

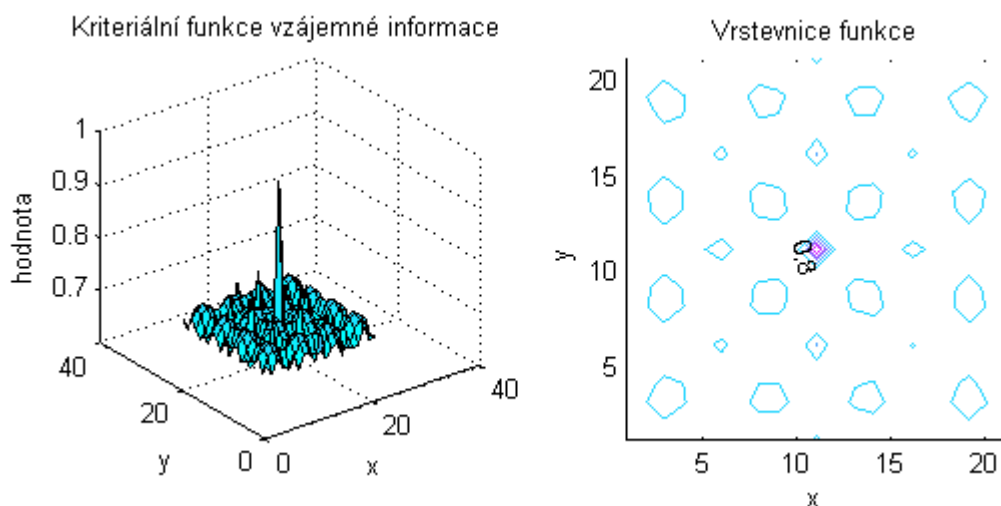


**Obrázek 10 Graf kriteriaální funkce (vlevo) a vrstevnice (vpravo) vzájemné informace vybrané části identických testovacích obrázců ve formě pruhů**



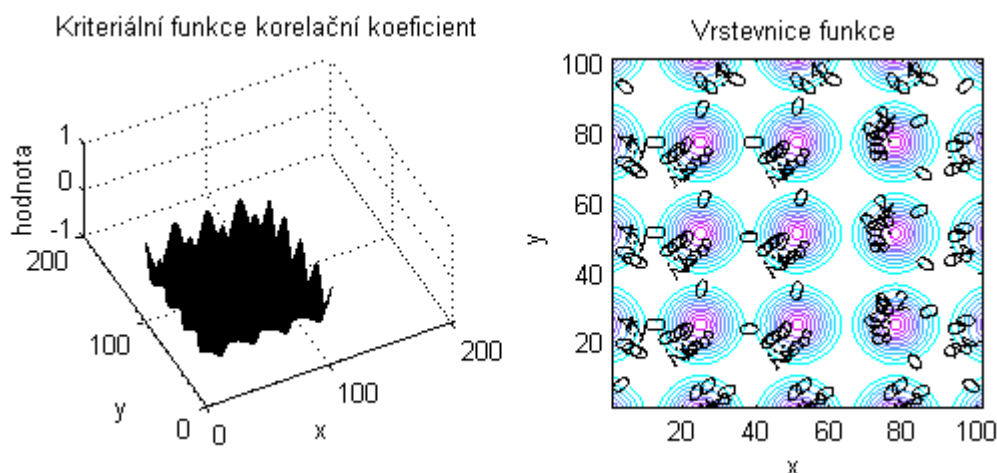


**Obrázek 11** Testovací obrazec složený z rozmazaných teček (vlevo) a jeho maska (vpravo)



**Obrázek 12** Graf kriteriální funkce (vlevo) a vrstevnic (vpravo) vzájemné informace vybrané části identických testovacích obrazců ve formě teček

Nepravidelný tvar vrstevnic funkce na Obrázku 12 (vpravo) je způsoben výběrem každého pátého pixelu rozmazaného obrazce. Důkazem je Obrázek 13, který obsahuje pravidelné vrstevnice funkce korelačního koeficientu.



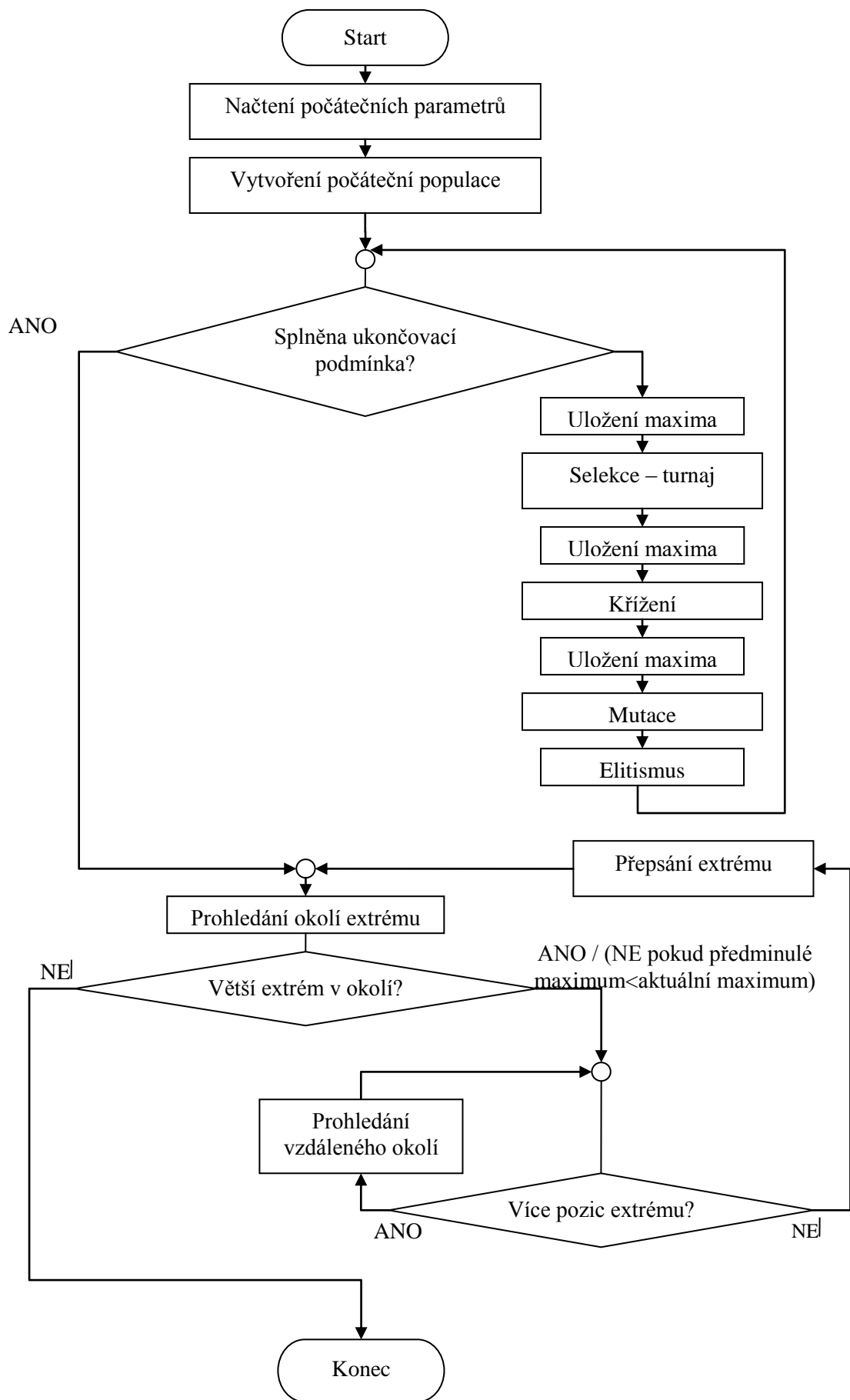
**Obrázek 13** Graf kriteriální funkce (vlevo) a vrstevnice (vpravo) korelačního koeficientu vybrané části identických testovacích obrazců ve formě teček; každý pixel

## 7 Realizace

V teoretické části jsme získali základní informace potřebné pro vytvoření programu pracujícího na bázi genetického algoritmu. Využívá se přitom návrhu z páté kapitoly. Realizace probíhá v programovém prostředí Matlab 2007b, jehož součástí je i grafické uživatelské prostředí (GUI).

### 7.1 Popis programu

Vytvořený program je upraven jednak pro optimalizaci testovacích funkcí, a jednak pro optimalizaci dvou obrazů, ze kterých se kriteriální funkce musí vypočítat pomocí registrace. Z důvodu testování a porovnávání kvality s jinými funkcemi je podstata obou algoritmů identická. Její průběh je znázorněn na stručném vývojovém diagramu Obrázku 14. První část se skládá z genetického algoritmu, jehož nejlepší výsledek optimalizace pokračuje do části s gradientní iterační metodou, u níž je cílem dosažení lokálního extrému. Tím se vytvořil hybridní genetický algoritmus, který stochasticky prochází různá řešení a detailně prohledává nejlepší dosažené.

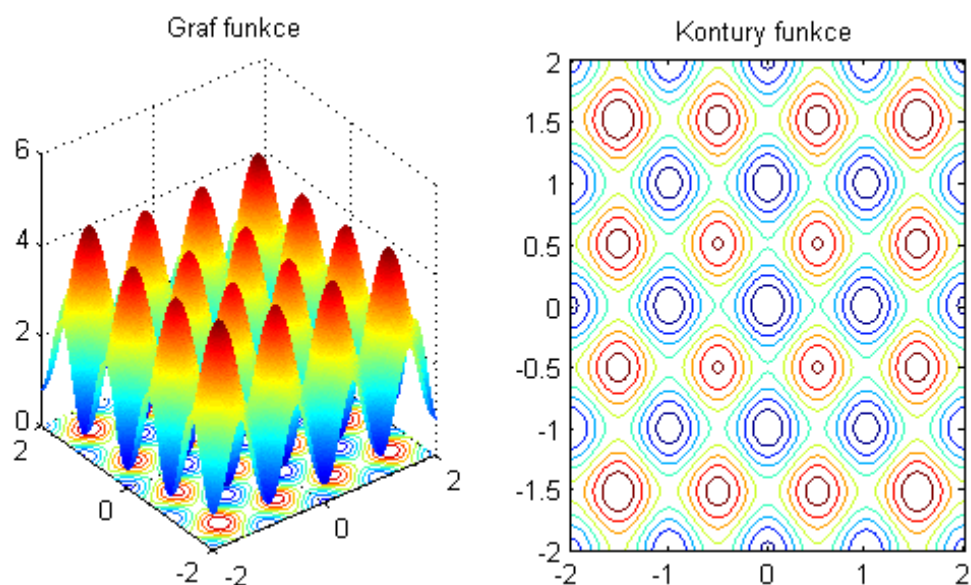


Obrázek 14 Vývojový diagram hybridního genetického algoritmu

## 7.2 Vytvořený genetický algoritmus

### Skript `Open_Optimalization_TestFunction.m`

Na začátku skriptu se provedly důležité úkony pro vyčištění paměti programového prostředí Matlab. Zabránilo se tak přepisu již vytvořených proměnných či ovlivňování proměnných v průběhu algoritmu. Příkaz `close all` vše zavřel, `clear all` vyčistil Workspace a `clc` vymazal všechny úkony v Command Windows, takže okno zůstalo prázdné. Po počátečních úpravách se nastavily okrajové hodnoty pro dva na sebe vzájemně kolmé směry a krok znázorňující vzdálenost dvou paralelních čar. Tím se vytvořila pravoúhlá síť připravená pro doplnění vypočítanými hodnotami z funkcí. Skript obsahuje nabídku šesti různých testovacích funkcí, jako je například Rastriginova, Ackleyho či Rosenbrockova funkce. Některé funkce obsahují drobné úpravy, což způsobí jejich posun v ose  $y$  nebo případnou změnu průběhu funkce. Podstata bude vždy spočívat v nalezení jednoho globálního minima. Pro znázornění průběhu funkce se její hodnoty zanesly do pravoúhlé sítě. Jelikož vykreslení samotného grafu může být dosti nepřehledné, nechaly se vykreslit i kontury funkce, které dobře znázorňují gradient, viz Obrázek 15.



**Obrázek 15 Graf Rastriginovy funkce v oblasti o rozměrech  $x,y \in [-2, 2]$  vlevo; kontury dané funkce vpravo**

Aby se nezanášela do testování časové náročnosti algoritmu zbytečná chyba z důvodu vykreslování grafů, byl časovač zapnut těsně před volbou počátečních parametrů dané metody optimalizace. Vstupními hodnotami do funkce hybridního genetického algoritmu se stala zvolená testovací funkce, velikost počáteční populace, počet iterací cyklu a dvě hodnoty určující prohledávaný 2D prostor. Na výstupu se očekává pozice a hodnota

nalezeného extrému. Závěr algoritmu patří vypsání získaných údajů a ukončení čítač pro zobrazení časové náročnosti algoritmu.

## **Funkce TestFunction.m**

TestFunction.m je název hlavní funkce, která obsahuje hybridní genetický algoritmus. Jejím vstupem, jak jsme již uvedli v předchozím odstavci, je zvolená testovací funkce, velikost počáteční populace, počet iterací cyklu a dvě hodnoty určující prohledávaný 2D prostor. Výstupem je též zmiňovaná pozice a hodnota extrému. K dispozici jsou pouze dvě čísla určující okrajové hodnoty funkce. Z důvodu transformace se rozepíší do čtyř hodnot určujících počátek a konec prohledávaného prostoru.

Nyní jsou již zadány všechny potřebné parametry, a tak se může přistoupit k vytvoření počáteční populace. V cyklu s ukončující podmínkou dovršením velikosti počáteční populace se náhodně generují hodnoty alel pro posun v ose x, posun v ose y a rotaci. Jelikož je algoritmus upravený pro porovnávání s jinými testovacími funkcemi, ve kterých se rotace neuzívá, je zde nastavena rotace na nulovou hodnotu. Údaje z hodnocení vzniklých chromosomů se uloží do proměnné data, která bude použita pro vytvoření závěrečných grafů.

Zbýlý genetický algoritmus je součástí velkého cyklu, jehož výstupem je chromosom s nejlepším ohodnocením. Ukončení cyklu nastane při dosažení nastaveného maximálního počtu iterací. I když algoritmus může dospět k hledanému extrému dříve, kvůli neznalosti hodnoty extrému není ukončovací podmínka v podobě dosažení určité hodnoty

Před změnou chromosomů se uloží nejlépe ohodnocený do nových proměnných (`posun_x_nej`, `posun_y_nej`), aby náhodnými výběry operátorů nenastala jeho změna, což by vedlo k odpoutání od případného správného řešení.

Vytvoření nové populace začíná selekcí, kdy se z aktuální populace vyberou jen některé rodičovské chromosomy. Z neměnné velikosti populace vyplývá, že určitý počet jedinců musí být zastoupeni vícekrát. Jako způsob selekce se zvolil turnaj, neboť jak bylo zmíněno v kapitole 1.4, zabývající se problémy registrace digitálního obrazu, turnaj patří mezi metody s nejnižší časovou náročností. Vytvořen byl jako čtyřnásobný cyklus, ve kterém se vygenerují dvě náhodná čísla a upraví se tak, aby jako celá čísla ležela v mezích velikosti populace. Přitom se bere v úvahu, že tyto dvě hodnoty nesmí být stejné, neboť by nemohlo dojít k souboji. V této situaci se pro jedno vybrané číslo zvolila jeho vedlejší pozice. Pokud se bude jednat o okrajovou hodnotu, je zajištěno posunutí směrem k ostatním hodnotám, aby nedošlo k překročení velikosti populace. Tyto dvě hodnoty označují pořadí vybraných chromosomů určených k souboji. Souboj probíhá porovnáním ohodnocení daných dvou chromosomů a vybráním většího, jehož hodnota se vloží do nového vektoru. Při situaci identického ohodnocení obou chromosomů se již výpočet dále zbytečně nekomplikuje a vybírá se první chromosom. Jelikož probíhal pouze výběr hodnot a nemohla vzniknout

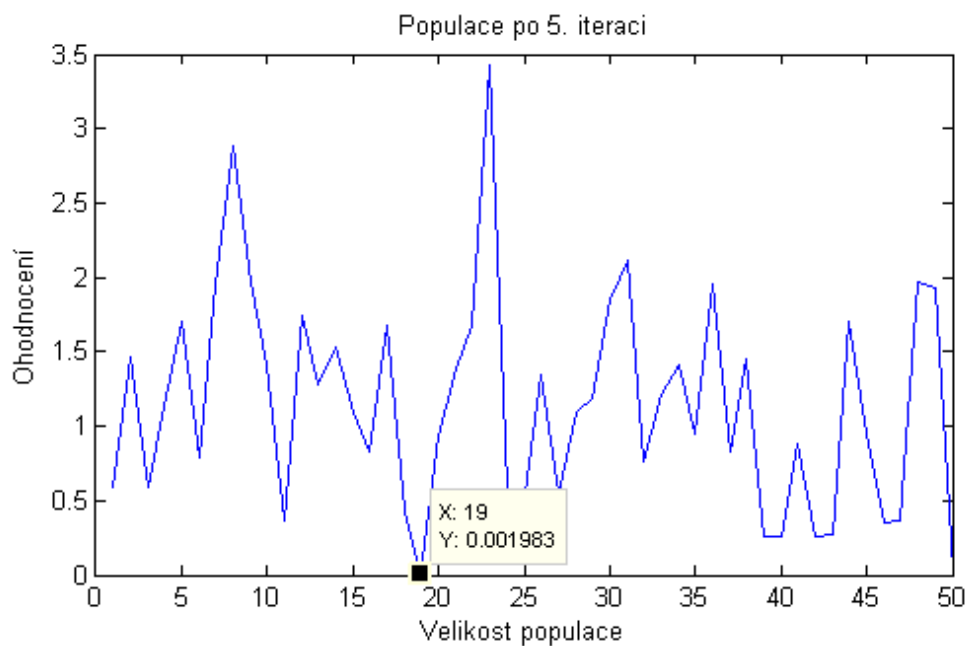
žádná lepší hodnota, odpadá tedy nutnost prozkoumat populaci za účelem zjištění nového extrému a přepisu původního.

Genetický algoritmus pokračuje aplikací operátorů, s čímž souvisí vytváření nových chromosomů. Po nastavení parametrů pravděpodobnosti křížení a pravděpodobnosti mutace se pokračuje samotným algoritmem křížení chromosomů. Náhodně se vygeneruje vektor jedniček a nul s délkou velikosti populace. Zastoupení jedniček je dáno pravděpodobností křížení. Pozice jedniček určují chromosomy, které podlehnou křížení. Pro lepší práci se vytvořily vektory hodnot na místech křížení a vektory hodnot bez změny křížením. Během cyklu se pro každý křížící se chromosom vygeneruje náhodné číslo udávající pořadí chromosomu vstupujícího s ním do operace, která vytvoří průměr alelových hodnot dvou chromosomů.

Jelikož ale ne všechny alely byly podrobeny operátoru křížení, vytvoří se výstupní populace obsahující jak nové chromosomy vzniklé zkřížením, tak chromosomy, které nebyly ke křížení předurčeny. Po zjištění ohodnocení upravené populace se provedla kontrola, zda se operacemi průměru nevytvořil nový extrém, který by musel být uložen pro pozdější operaci elitářství.

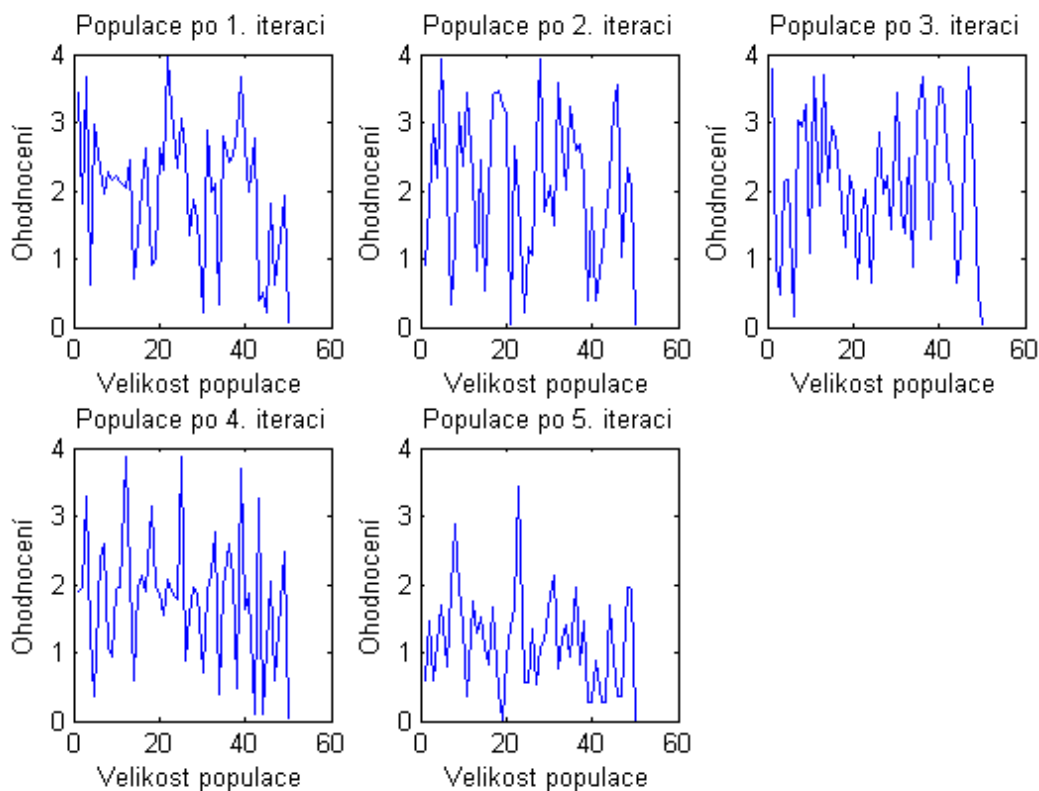
Další úpravy chromosomů mohou nastat při mutaci. Ta je založená stejně jako křížení na náhodném vygenerování vektoru nul a jedniček s délkou velikosti populace, přičemž z důvodu navoleného malého čísla pravděpodobnosti mutace bude míst s hodnotou jedna podstatně méně, než pozic s hodnotou nula. Opět proběhne v závislosti na vygenerovaném vektoru rozepsání alel do dvou skupin podle aktivace k mutaci. V případě nulové mutace všech chromosomů se pouze přepíšou názvy alel. V druhém případě se pro každou alelu mutujícího chromosomu náhodně vygeneruje nová hodnota alely a opět dojde ke sloučení mutovaných a nemutovaných chromosomů.

Následuje elitismus, ve kterém se do populace přidá na základě ohodnocení nejlepší chromosom z celé iterace a odebere se nejhorší. Takto upravená populace se může vykreslit, jako je tomu na Obrázku 16, z kterého lze číst velikost populace o padesáti jedincích s rozsahem ohodnocení od 0,002 do 3,4 jednotek. Algoritmus je upraven pro porovnávání s jinými testovacími funkcemi, takže se hledá globální minimum funkce. Nejsilnější chromosom s hodnotou 0,001983 se nalézá na devatenáctém a padesátém místě grafu. Uspořádání jednotlivých chromosomů jsou nahodilá, pouze poslední chromosom je spojen s elitismem, tedy signalizuje maximální hodnotu dosaženou během celé iterace.



**Obrázek 16 Ohodnocení chromosomů v populaci po průběhu páté iterace**

Vykreslením všech grafů populací v jednom okně se zpřehledňuje vývoj kvality populací, jako je tomu na Obrázku 17. Je však umožněno vykreslení maximálně dvanácti iterací, neboť větší počet grafů by snižoval přehlednost a čitelnost dat.



**Obrázek 17 Graf vývoje kvality populace během pěti iterací**

Na závěr se z konečné populace vybere jedinec s nejlepším ohodnocením, které poukazuje na nejlepší nalezenou pozici optimalizace. Jelikož se však touto metodou téměř nikdy nedosáhlo lokálního extrému, vytvořila se iterační gradientní metoda, čímž se genetický algoritmus stal hybridním genetickým algoritmem.

Iterační gradientní metoda je postavena na hodnocení okolí a posouvání se proti směru růstu gradientu. Celý výpočet se provádí v cyklu, jehož ukončovací podmínka spočívá ve změně hodnoty pro proměnnou *konec*, či dosažení nastavených hraničních hodnot funkce. Velikost prohledávaného okolí si uživatel volí buď čtyř-hodnotové, zaujímavější tvar kříže, nebo osmi-hodnotového s případným osmi-hodnotovým okolím ve vzdálenosti 4 kroků od hledané hodnoty. Nejlépe se osvědčilo osmi-hodnotové, protože v případě čtyř-hodnotového se na šikmý směr muselo provádět stejné množství iterací, a přitom se mohly některé hodnoty opomenout, a v případě šestnácti-hodnotového byl výpočet příliš náročný. Vždy se začíná prohledávat vpravo a postupuje se po směru hodinových ručiček, takže v případě osmi-hodnotového okolí lze směr vyjádřit pomocí světových stran jako V-JV-J-JZ-Z-SZ-S-SV. Po zjištění hodnot v okolí aktuálního extrému se všechny hodnoty porovnají. Pokud se v okolí našel jeden větší extrém, přepíše se pozice i hodnota minulého extrému a cyklus se opakuje. Při nalezení více extrémů nastává složitější situace, kdy se začnou zjišťovat hodnoty vzdálenějšího okolí do situace, dokud se nenalezne pouze jedno maximum. Vzdálenost okolí však může narůstat pouze do 20 kroků, aby se vyhnulo uvážnutí. Případ pouze nižších ohodnocení v okolí ještě nesplňuje podmínku ukončení zpřesňující metody, neboť se toleruje jedna možnost neočekávaného šumu, a tak pro ukončení celého cyklu je zapotřebí získat nižší hodnotu i ve vzdálenějším okolí. Tato úprava byla aplikována po testování optimalizace dvou snímků, ve kterém se zjistili mnohem lepší výsledky optimalizace. Podmínka průběhu prohledávání okolí obsahuje více výroků a typů logických spojek. Symbolický zápis je ve tvaru:

$$(A \vee B) \wedge C$$

kde A představuje výrok ‚Lepší hodnota v okolí‘, B výrok ‚Prohledání jen blízkého okolí‘ a C výrok ‚Aktuální extrémní ohodnocení je různé od předminulého extrémního ohodnocení‘.

Řešení složeného výroku je obsaženo v Tabulce 3.



**Tabulka 3 Řešení složeného výroku (A∨B)∧C**

Lepší hodnota v okolí	Prohledání jen blízkého okolí	Aktuální extrémní ohodnocení je různé od předminulého extrémního ohodnocení	Řešení situace
1	1	1	If, nahrazení novým ohodnocením, pokračování v cyklu s výrokem B=1
1	1	0	Nenastane – 1. zdůvodnění
1	0	1	If, nahrazení novým ohodnocením, pokračování v cyklu s výrokem B=1
1	0	0	Elseif, nahrazení novým ohodnocením, konec cyklu
0	1	1	If, minulé ohodnocení, pokračování v cyklu s výrokem B=0
0	1	0	Else, minulé ohodnocení, konec cyklu
0	0	1	Else, minulé ohodnocení, konec cyklu
0	0	0	Nenastane – 2. zdůvodnění

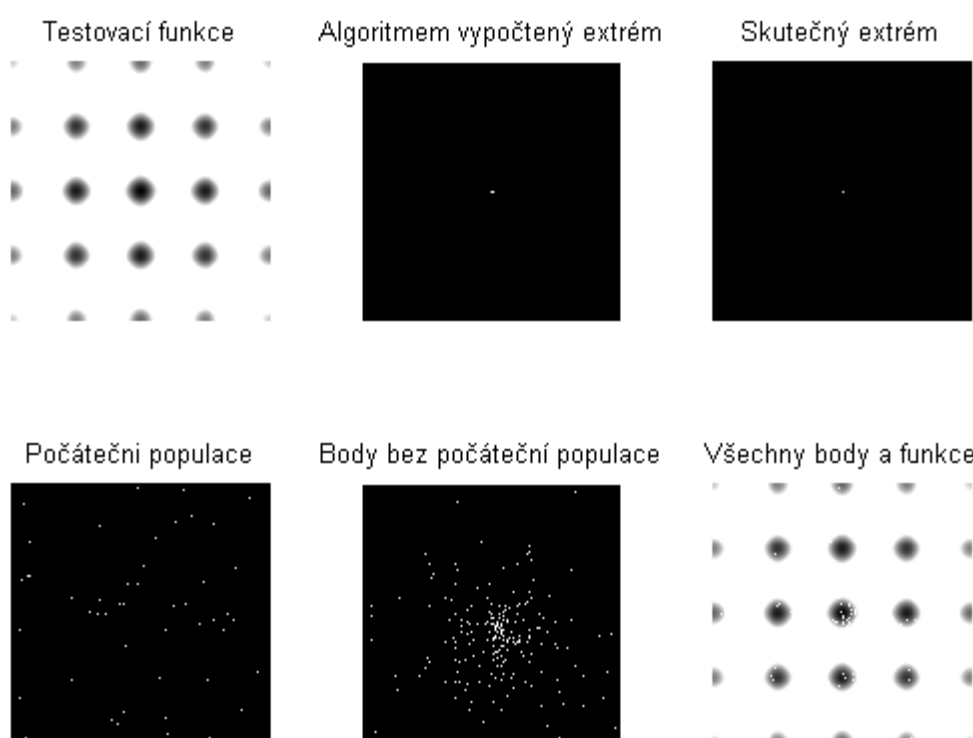
Pokud je výrok ‚Prohledávání jen blízkého okolí‘ nastaven na nulu, předminulý extrém musel být větší než minulý, např. ohodnocení = {2 1,5 aktuální extrém}.

První zdůvodnění: Příklad A=1,B=1 a C=0 nikdy nemůže nastat, neboť podle prvního výroku by musela být hodnota extrému lepší než minulá, např. ohodnocení = {1,5 2}, podle druhého výroku by předminulá hodnota musela být ještě nižší, např. ohodnocení = {1 1,5 2}, což by bylo v rozporu s třetím výrokem, který v tomto případě požaduje aktuální a předminulou hodnotu stejnou, tudíž ohodnocení = {1 1,5 1} nebo ohodnocení = {2 1,5 2}.

Druhé zdůvodnění: Příklad A=0,B=0 a C=0 také nikdy nemůže nastat, protože se opět vylučují hodnoty druhého a třetího výroku. Příklad prvního výrok je ohodnocení = {2 1}, přidáním druhého vznikne ohodnocení = {3 2 1}, což opět koliduje s požadavkem třetího výroku, aby aktuální a předminulý extrém byl identický.

Pro vizuální porovnání dosud získaných výsledků si v jednom okně vykreslíme sadu podoken, viz Obrázek 18. V prvním je znázorněna použitá testovací funkce v dvourozměrném prostoru. Lze na ní vidět lokální minima a maxima, která patří Rastriginově testovací funkci pro  $x,y = \langle -2, 2 \rangle$ . Na obrázku umístěném v horní řadě uprostřed je vyznačen jeden světlý bod, který znázorňuje nalezený extrém funkce pomocí hybridního genetického algoritmu. Pro porovnání se uvádí skutečný extrém funkce, zobrazený na obrázku v horní řadě vpravo. Vizuálním porovnáním či použitím datového kursoru obrázku zjistíme správnost navrženého algoritmu. Pokud nebude viděn na daných

dvou obrázcích žádný extrém, je zapotřebí zvětšit rozsah jeho pozice, neboť je pravděpodobně zvolena velká plocha funkce, která zapříčiní podvzorkování nalezeného bodu. Na obrázku spodní řady vlevo jsou pomocí bílých teček vyznačeny chromosomy počáteční populace. Přibližně rovnoměrné rozprostření vypovídá o náhodném procesu výběru alel počátečních chromosomů. Při vykreslení populací vzniklých iteracemi se rozprostření již řídí hlavně selekcí nejsilnějších jedinců a operátorem křížení, takže hustota chromosomů se začíná soustřeďovat do oblastí nejlepšího ohodnocení. Počet chromosomů se již nedá určit, neboť algoritmus není vybaven proměnnou, zaznamenávající selekci nevybrané či křížením a mutací nově vzniklé chromosomy. Znalost této proměnné není podstatná. Na posledním obrázku spodní řady vpravo se nachází informace o testovací funkci a všech vzniklých jedincích, takže uživatel může vizuálně zhodnotit průběh a kvalitu algoritmu. Pokud se extrém nenalezne správně, uživateli se naskytují informace o příčinách nezdaru, jakými je například příliš malý počet chromosomů v populaci na příliš velkou prohledávací plochu funkce.



**Obrázek 18 Závěrečné vykreslení; vrchní řada: půdorys testovací funkce, algoritmem vypočtený extrém, skutečný extrém; spodní řada: chromosomy počáteční populace, chromosomy během všech iterací, všechny chromosomy zobrazené na testovací funkci**

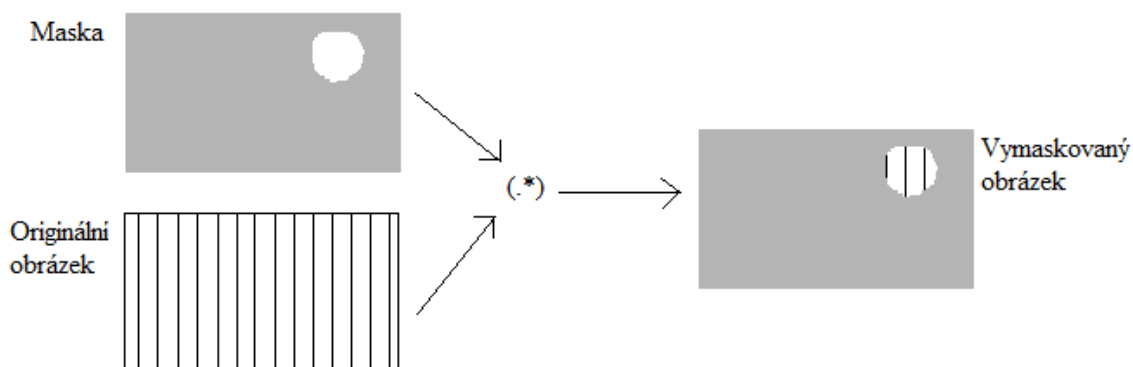
Závěr algoritmu poskytuje informaci o ujištění, zda doopravdy neexistuje u použitých jedinců lepší ohodnocení, než jaké bylo zvoleno.

## Funkce GeneticAlgorithm.m

Jak se již uvedlo, funkce GeneticAlgorithm je svojí podstatou stejná jako funkce TestFunction. Zásadní rozdíl spočívá v přizpůsobení prvního algoritmu práci s obrazy a druhého algoritmu testovacím funkcím. Nyní se popíší pouze úseky, od kterých se GeneticAlgorithm liší od TestFunction.

Vstupními parametry funkce GeneticAlgorithm jsou dva obrazy, u nichž se bude zjišťovat optimalizace, maska, vektor minimálních hodnot alel, vektor maximálních hodnot alel, údaj o velikosti počáteční populace a číslo s počtem průběhů iterací. Ve výstupu algoritmu nalézáme údaj o nejsilnějším chromosomu, tedy vektor s nejlépe hodnocenými alelami pro posuny a rotaci. Druhým údajem výstupu není pouze nejlepší hodnota optimalizace ale matice hodnot obrazu, podrobena nejlepšímu jedinci. Tento údaj je zvolen z důvodu vykreslení optimalizace v grafickém prostředí, kde znalost dosaženého maximálního ohodnocení není podstatná.

Výraznější změna od algoritmu testovací funkce nastává při vytváření počáteční populace, kdy se hledají takové chromosomy, které by zaručily existenci celé masky v obou obrazech. Maskou se rozumí matice jedniček a nul, s jejíž pomocí se vyznačí oblast referenčního obrazu jako na Obrázku 19.



**Obrázek 19** Vytvoření ‚vymaskovaného‘ obrazu [30]

Takto upravený (vymaskovaný) referenční obraz vstupuje do registrace s plovoucím obrazem, a jako výsledek se získává hodnota dané registrace, neboli ohodnocení chromosomu.

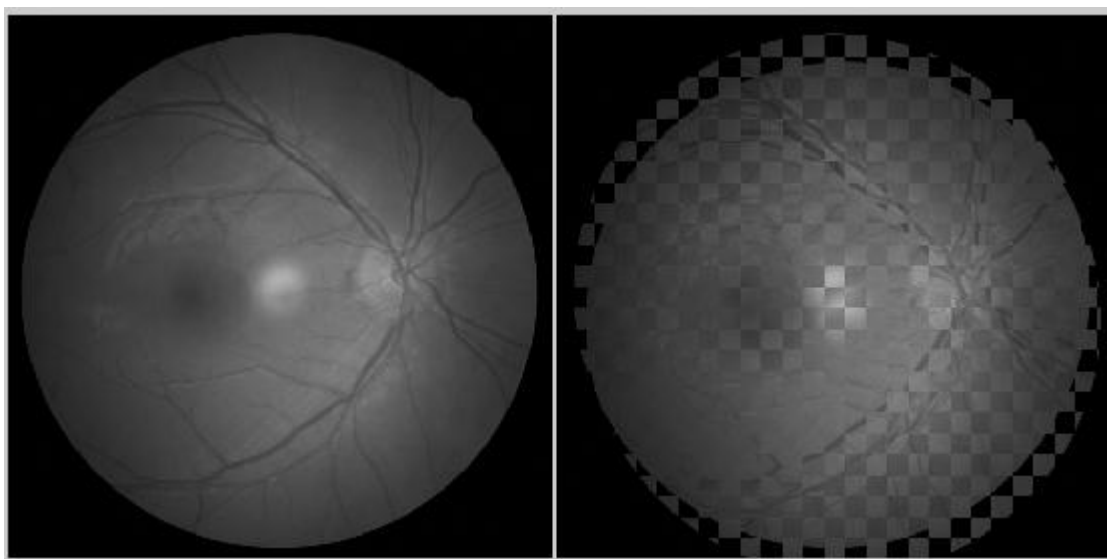
V algoritmu je situace vytváření počáteční populace řešena náhodnou volbou alel, které jsou podrobny funkci Register. Výsledkem funkce je zjištění ohodnocení a případný impuls pro opakování cyklu za účelem vytvoření dalšího jedince.

Další obměna testovacího algoritmu se objevuje při vytváření křížení a mutace chromosomů, křížení dvou nejsilnějších jedinců a během gradientní metody. Tedy

v případech, kdy potřebujeme vytvořit ohodnocení pro danou transformaci obrazu, neboť nám není známo. Opět se tedy volá funkce Register.

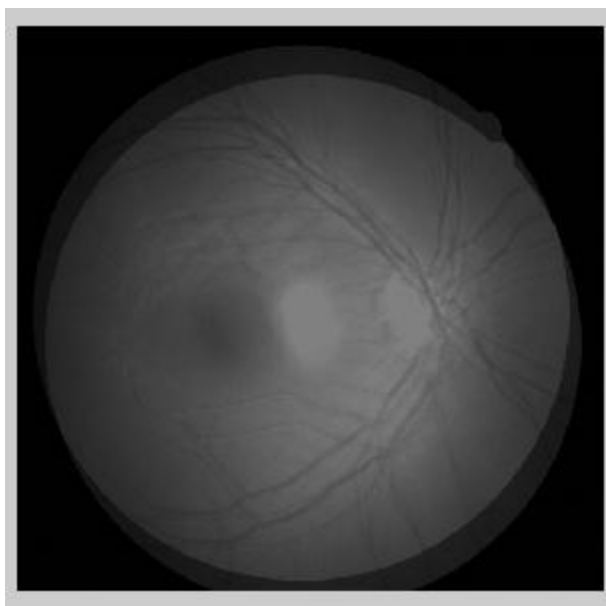
Jelikož nebudeme znát přesný průběh celé ohodnocující funkce, nebudeme vytvářet obrazy znázorňující chromosomy. Z tohoto důvodu není potřeba zaznamenávat hodnoty všech chromosomů do proměnné data, jako tomu bylo u funkce TestFunction.

Výsledek optimalizace se vykreslí šachovnicí a překrytím obrazů. V obou typech se využívá referenční obraz a obraz s nejlépe ohodnocenou transformací. Správným překrytím se dokazuje kvalita optimalizace. Při vytvoření šachovnice se využívá funkce checkerboard obsažená v Image Processing Toolboxu. Příkaz `checkerboard(n, rady_sach, sloupy_sach) > 0` vytváří čtvercovou šachovnici, kde proměnná `n` nese informaci o počtu pixelů po stranách každého čtverce, `rady_sach` informují o počtu řad a parametr `sloupy_sach` o počtu sloupců. Pokud je příkaz větší než nula, vytvoří se šachovnice pouze s bílými a černými poli. Tím poslouží jako maska referenčního obrazu a inverzní maska u výstupního obrazu optimalizace. Při spojení těchto dvou sad dat vzniká šachovnicový obraz, ve kterém nesmí být poznat přechody ve spojích, jak o tom pojednává Obrázek 20.



**Obrázek 20 Šachovnicový způsob vizualizace výsledku optimalizace dvou identických obrazů; vlevo dokonalá, vpravo nedokonalá s hodnotou horizontálního posunu 11 a vertikálního -27 pixelů**

Druhým způsobem pro vizuální ověření optimalizace je vytvoření zprůměrnovaného snímku. Využije se opět obraz referenční s obrazem podrobeným optimalizaci. Výhoda tohoto typu spočívá ve zviditelnění všech struktur z obou obrazů, takže názorně vidíme překrývající se části.



**Obrázek 21 Zprůměrovaný obraz tvořený referenčním a špatně zregistrovaným s hodnotou horizontálního posunu 11 a vertikálním -27 pixelů**

### **Funkce Register.m**

Tato pomocná funkce v sobě zahrnuje transformaci registrovaného obrazu a registraci takto pozměněného obrazu s ‚vymaskovaným‘ referenčním obrazem. Jejimi vstupy jsou tedy dvě sady dat představující obrazy, vektor hodnot určující transformaci, maska a proměnná `uspesne_posunuti`, která obsahuje logickou hodnotu o stejné ploše masky v obou obrazech. Pokud se tedy tato proměnná během průběhu algoritmu Register změní je zaručeno zmenšení cyklu o jednu iteraci, neboť proběhla úspěšná registrace obrazů za vzniku nového chromosomu. Výstupem funkce je ohodnocení vektoru pro transformaci a opět proměnná `uspesne_posunuti`.

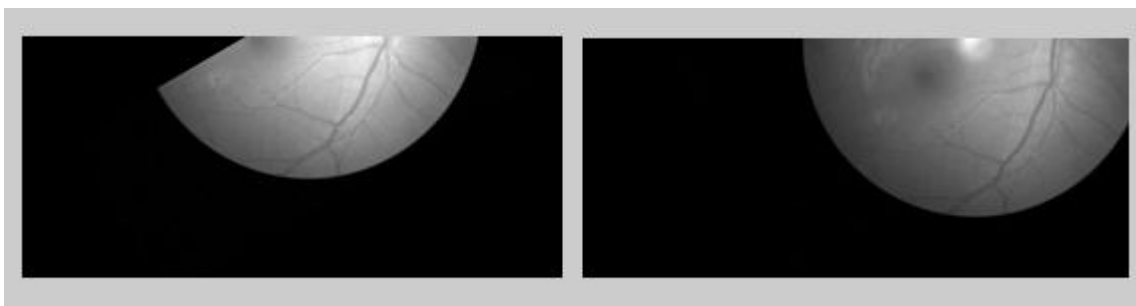
Na začátku se vytvoří proměnná `konstantni_obr2` o velikosti plovoucího obrazu obsahující pouze jedničky. Bude potřebná při kontrole masky v plovoucím obraze. Další proměnná s názvem `vymaskovany_obrazek1` vznikne násobením referenčního obrazu s maskou převedenou z binárních do osmi-bitových čísel.

Samotná transformace probíhá ve funkci `Transformation`, ze které se jako výstup získá transformovaný obraz v závislosti na hodnotách vektoru posunů a rotace. Se stejnými hodnotami vektoru se provede i transformace proměnné `konstantni_obr2`, který po ‚vymaskování‘ změní název na `overeni_plochy2`. Porovnáním obsahu hodnot o velikosti jedničky v maticích `overeni_plochy2` a maska se určí, zda se celá maska objevuje v obou obrazech. Pokud by tato podmínka nebyla naplněna, daná transformace by byla považována za neplatnou a muselo by dojít k opakování cyklu. Podmínka však ve skutečnosti není úplně přesná z důvodu digitalizace a interpolace hodnot při použití rotace. Může totiž docházet ke ztrátě nebo nechtěnému vytvoření hodnot na krajních pixelech

obrazu. Při splnění podmínky se pomocí funkce MI2 získá ohodnocení dané registrace a změní se hodnota proměnné `uspesne_posunuti`. Praktické využití této proměnné je při práci s obrazy, ve které se určuje maska. Při práci s vytvořenými funkcemi je hodnota úspěšného posunutí automaticky přepsána vždy, neboť se vybírají pouze hodnoty ve zvolených rozměrech funkce, takže nemůže dojít ke kolizi v podobě špatně zvolených hodnot alel.

## Funkce Transformation.m

Základ pro vytvoření funkce Transformation jsem převzala ze své bakalářské práce, která se zabývala porovnáním kritériálních funkcí. V této diplomové práci však byl algoritmus upraven, neboť posuny se do výpočtu zanášela chyba, která se projevila při použití rotace. Její Výsledek je zřejmý na Obrázku 22. Pravá část vznikla průběhem starého algoritmu, ve kterém se nejprve aplikovaly posuny. Ty způsobily posun obrázku pomocí ořezání části dat. Následné použití rotace sice způsobilo otočení, avšak v paměti scházela data celého snímku. Tento problém musel být tedy odstraněn, neboť na rozdíl od bakalářské práce se rotace zde bude využívat.



**Obrázek 22 Výsledek starého algoritmu transformace vlevo, výsledek zdokonaleného algoritmu transformace vpravo**

Mezi vstupní veličiny patří registrovaný obraz, hodnota vertikálního posunu, horizontálního posunu a rotace. Úprava starého algoritmu spočívá ve vytvoření dostatečně velkého okolí snímku o nulových hodnotách. Za tímto účelem se vypočítala délka úhlopříčky obrázku, jejíž délka se přičetla ke každé straně snímku. S naprostým vymaněním se z oblasti své původní pozice se nepočítá.

Nejprve se uvažovala transformace rotací, neboť její osa otáčení prochází středem obrázku. V případě nedodržení tohoto pořadí by vznikl zcela odlišný obrázek, protože by se rotací měnily i souřadnice posunů, viz Obrázek 22. K vytvoření transformace se využila funkce `imrotate` z Image Processing Toolboxu. Příkaz volání má tvar `imrotate(obr, uhel, 'bicubic', 'crop')`, kde se určuje objekt posunutí, úhel posunutí, metoda interpolace a možnost ořezání na původní velikost (`,crop'`). Pro interpolaci jsme zvolili bikubickou metodu, neboť dávala mnohem lepší výsledky než metoda nejbližších sousedů (`,nearest'`).

Funkce pokračuje dvěma cykly, kde podmínkami jsou velikost posunu  $x$  a posunu  $y$ . V případě záporného  $x$  se vymaže takový počet sloupců, kolik je určeno absolutní hodnotou z čísla  $x$ . Poté se zjistí rozměry zmenšeného snímku. Posunutý obraz se vytvoří sloučením zmenšeného obrazu a maticí nul o počtu sloupců stejným, jako odstraněných sloupcům, a o stejném počtu řad, jako je tomu u zmenšeného obrazu. Původní a nově vytvořený obraz mají tedy stejné rozměry, avšak hodnoty v novém obraze jsou posunuty doleva. Při nulové hodnotě posunu  $x$  se pouze přepíše název obrázku. Pokud bude hodnota  $x$  kladné číslo, situace se bude řešit podobně jako u kladného posunu  $x$ . Vymazány budou však hodnoty vpravo a matice nul se umístí před zmenšenou matici. Podobný postup bude použit i při vertikálním řešení.

Po průběhu všech transformací se obrazu vrátí jeho původní rozměr. Do proměnné se přepíše matice hodnot, které se nalézají na místě původního obrazu. Tím vznikl transformovaný obraz, který je výstupem funkce.

### **Funkce MI2.m**

Algoritmus je postaven na vzájemné informaci, která vytváří kritériální funkci pro dané hodnoty transformace. Funkce i s její vloženou funkcí `point_h.m` nám byla dodána již při řešení bakalářské práce, ve které si čtenář může zjistit podrobnější informace o popisu MI2. [30]

## **7.3 Grafické prostředí**

V praxi se spíše setkáváme s případem, kdy neznáme průběh funkce obsahující hledaný extrém. V našem případě se jedná o lícování dvou snímků. Kritériální funkce se vytvoří na základě transformace obrazu a jejího ohodnocení. Při aplikaci formou genetického algoritmu se tedy opět populace skládá z chromosomů, jejichž alely představují posun horizontální, posun vertikální a nyní se zapojí i alela pro rotaci. Podstatný rozdíl je v zavedení tvorby masky, neboť podrobit všechny body obrazu registraci za účelem zjištění ohodnocení chromosomu by bylo časově i výpočetně velmi náročné.

Matlab nabízí nástroj pro vytvoření programu s uživatelsky příjemným prostředím. Dotyčný tedy nemusí být znalý programování, aby mohl použít výpočtu implementovaného algoritmu. Do styku přijde jen s tlačítky obsahujícími příkazy akce, kolonkami pro vyplnění proměnných a okny pro vykreslení grafů nebo obrázků. Vše musí být přehledné a účelné.

### **Algoritmus GUI – funkce `Open_Optimization`**

K vytvoření grafického prostředí jsme využili LayOut Editoru. V tomto novém okně jsme si rozvrhli a nastavili jednotlivé prvky. Uložením se v Editoru Matlabu vytvořil nový

M-file. Obsahoval základní struktury většiny vytvořených prvků v podobě funkcí. Naším úkolem bylo správné propojení funkcí a vyplnění jejich struktury potřebným algoritmem.

Nejprve jsme si vytvořili zcela novou funkci, která se věnuje inicializaci proměnných při otevření uživatelského prostředí. Automaticky se načtou do tohoto prostředí předvolené hodnoty.

Vnitřní algoritmus funkcí pro číselné kolonky, jako je například zadání velikosti počáteční populace, se rozepsal jako převod textu na číslo, se kterým se dál bude pracovat ve formě proměnné. Pokud bude do kolonky vepsán text a ne číselná hodnota, vytvoří se chybová hláška. Jedinou kolonkou bez požadavku číselné hodnoty je název vytvořené masky.

V grafickém prostředí uživatel nejprve použije tlačítko ‚Vyber obrázky‘. Příkazem `uigetfile` (`{ '*.jpg; *.png'; '*.m'; '*.*' }`, ‚Výběr souboru‘) se vytvoří dialogové okno s názvem Výběr souboru, které je cestou k cílovému souboru. Informace ve složené závorce informují o typech souborů, které lze načíst. Upřednostňují se soubory ve formátu jpg a png, které se v dialogovém okně nabídnou jako první. Vybraná data jsou přes příkaz `imread` uložena do globální proměnné `obrazek1`. Stejným způsobem se vytvořila i globální proměnná obsahující druhou sadu dat. Pro přenos dat do grafu byla funkce propojena s LayOut Editorem přes příkaz `axes`. Z důvodu nepraktičnosti rgb obrázků se převod na osmi-bitové obrazy provádí již zde. Dokončení průběhu funkce nastaví změnu aktivace pro tlačítka načtení a vytvoření masky.

Ve funkci tlačítka ‚Vytvořit masku s názvem:‘ jsou definovány globální proměnné `obrazek1` a `maska`. Po zavolání proměnné o velikosti převzorkování se upraví velikost referenčního obrazu. Přes příkaz `roipoly(obrazek1)` se načte převzorkovaný referenční obraz a uživatel si zvolí oblast zájmu. Takto vzniklá maska se uloží. Volbou tlačítka ‚Načíst masku‘ se volá již vytvořená maska stejným způsobem, jako tomu bylo u vstupních obrazů a ukládá se do globální proměnné `maska`. Ať se již vybere jakákoli možnost získání masky, druhé nevybrané tlačítko se deaktivuje a u posledního tlačítka s názvem ‚Spust’ výpočet‘ se nastaví aktivní stav. Aktivace zaručí získání všech potřebných informací pro hladký průběh algoritmu.

Funkce posledního tlačítka ‚Spust’ výpočet‘ v sobě zahrnuje celý algoritmus optimalizace. V této funkci jsou rozepsány pouze počáteční parametry a hodnoty výstupů, neboť zbytek algoritmu je obsažen ve vložené funkci `GeneticAlgorithm`. Po načtení globálních proměnných se zapne časovač. U vytváření rozostření se využívá nabídky Image Processing Toolboxu, odkud se převzal příkaz `fspecial('average', hsize)`. První parametr závorky představuje typ použitého filtru, kterým se vytváří rozostření, a `hsize` představuje velikost matice. Takto vzniklý 2D filtr je použit na obrázek pomocí příkazu `imfilter`.



Případná změna převzorkování se provede přes příkaz `imresize(obrazek1, scale)`, kdy pro hodnotu `scale > 1` vznikne převzorkovaný obraz a v opačném případě podvzorkovaný.

Nyní se již zavolá funkce `GeneticAlgorithm`, jejíž výsledky se jednak vykreslí do grafu v podobě šachovnice, a jednak zapíše do textových polí v hlavním dialogovém okně.

Je nutno podotknout, že před spuštěním celého algoritmu optimalizace musí být připravený prostor pro plnění proměnných. Proto se vytvořil skript obsahující tři příkazy na zavření oken, vyčištění a paměti i obrazovky. Dále už obsahuje pouze odkaz na funkci GUI. Tento skript se najde pod názvem `Open_Clear_Optimization`.

### **Vytvořené grafické prostředí algoritmu**

Vytvořená funkce `GeneticAlgorithm` obsahuje spoustu možností výběru parametrů. Uživateli je tedy dána možnost zvolit si vlastní hodnoty parametrů podle aktuálních potřeb. Grafické prostředí jsme navrhli pro volbu parametrů v levé části okna a vytvoření grafů s výslednými hodnotami v pravé části, viz Příloha B.

Volba a načítání parametrů se řídí podle umístění jednotlivých příkazů. Uživatel tedy postupuje z levého horního rohu k levému dolnímu rohu, Obrázek 23. Po stisku tlačítka ‚Vyber obrázky‘ se zobrazí dialogové okno s nabídkou procházení složek v počítači za účelem zvolení sady vstupních dat. Po postupném vybrání obou obrazů se vyplní v hlavním dialogovém oknu prostor vpravo nahoře originálními obrazy a pod nimi se načte údaj o velikosti dat. Tento údaj slouží uživateli ke korekci velikosti dat pomocí změny hodnoty vzorkování. Načtení snímků aktivovalo dvě tlačítka pro vytvoření či zavolání masky. Výběrem příkazu ‚Načíst masku‘ opět můžeme procházet složky a vybrat již uloženou masku. Rozměry masky však musí odpovídat upraveným rozměrům načtených obrazů. Při volbě tlačítka ‚Vytvořit masku s názvem:‘ se nejprve musí napsat do rámečku název masky, pod kterým budou data uložena a musí se zvolit velikost vzorkování v procentech. Po stisknutí tlačítka se zobrazí první načtený obraz, ve kterém se při použití kurzoru vybere oblast zájmu. Aktivuje se dvojklikem do vybrané uzavřené oblasti. Volí se s ohledem na velikost rozměrů obrazu, neboť bude přímo úměrná časové a výpočetní náročnosti. Může však být nepřímo úměrná přesnosti optimalizace. Svou roli bude hrát i struktura neboť při výběru homogenní oblasti se těžko bude hledat správný extrém.

Vyber obrázky

Maska

Vytvořit masku s názvem:  Načíst masku

.mat  
(bez diakritiky a mezer)  
Před vytvořením zadej vzorkování!

Volba parametrů

Velikost počáteční populace:

Počet iterací:

Meze posunu v ose x:

Meze posunu v ose y:

Meze úhlu:

Pravděpodobnost křížení:

Pravděpodobnost mutace:

Další možnosti

Vzorkování obrazů:  %

Rozostření

Žádné

Obou obrazů

Jednoho obrazu

Načtená data

Velikost načtených dat: /

Vymaskovaný referenční obraz

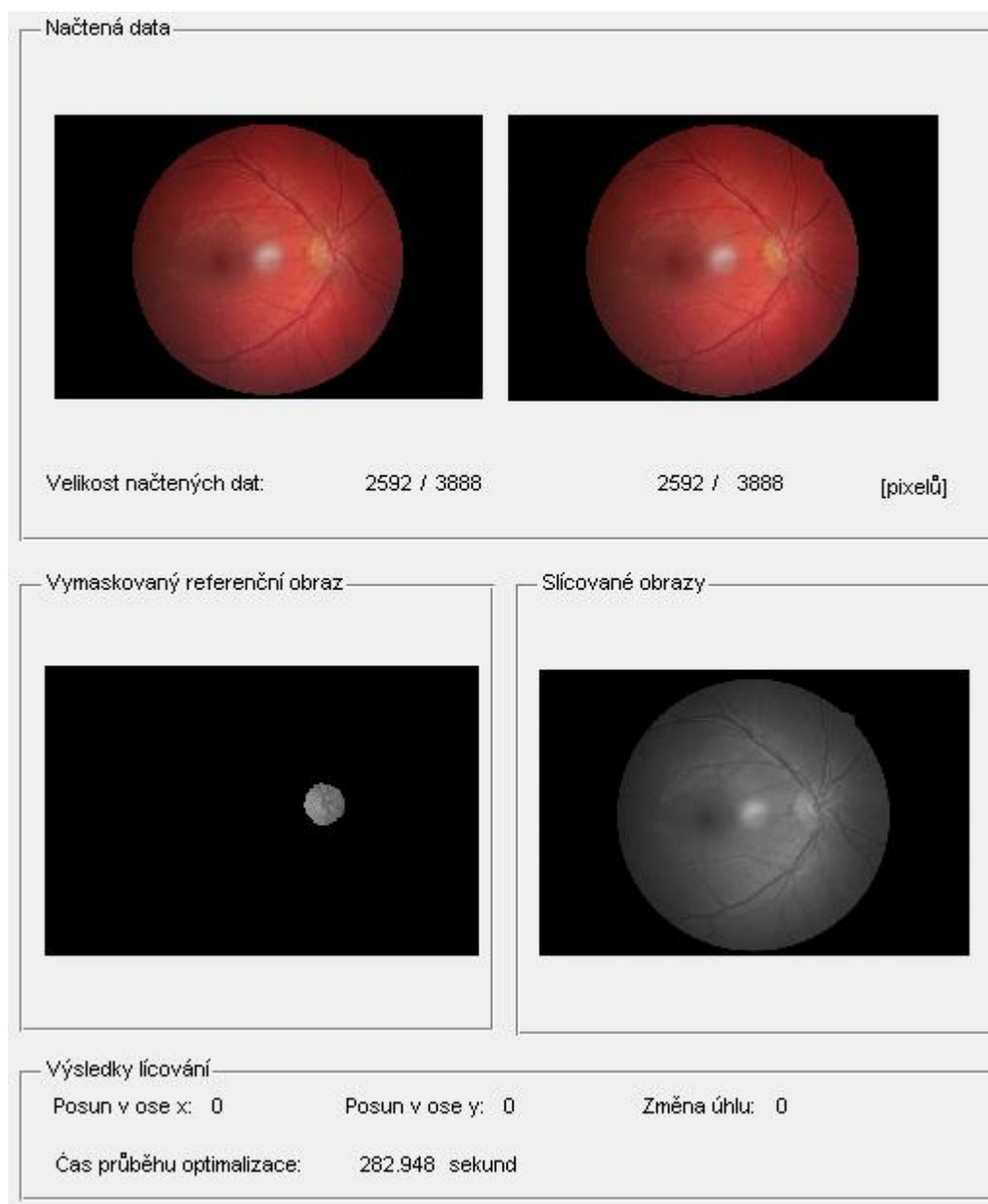
Výsledky lícování

Posun v ose x:                      Posun v ose y:

Čas průběhu optimalizace:                      sekund

**Obrázek 23** Levá část a polovina pravé části hlavního dialogového okna s příkazy a nastavenými počátečními parametry

Volba počátečních parametrů je v hlavním dialogovém okně již přednastavena na určité hodnoty. Uživatel si je však může libovolně měnit. Mezi další možnosti nastavení patří změna vzorkování a rozostření snímků. Hodnota vzorkování je přednastavena na 20%, což zmenší obraz na pětinu původní velikosti. Tím se umožní použití mnoha-pixelových obrazů pro jejich optimalizaci, u kterých by v případě nezmenšení dat byla optimalizace zdlouhavá. Při požadavku neměnné velikosti rozměrů se hodnota 20 přepíše na 100.



**Obrázek 24** Pravá polovina hlavního dialogového okna; ve vrchní řadě dva načtené vstupní snímky, ve prostřední řadě aktuální maska vlevo a slícované obrazy vpravo, ve spodní řadě výpis optimálních hodnot transformace a časové náročnosti průběhu

Posledním nastavením je volba rozostření. Panel nabízí možnost nulového rozostření, rozostření jednoho obrazu nebo rozostření obou obrazů. Rozostření jednoho obrazu simuluje situaci, ve které se provádí registrace na podobných datech. Při rozostření obou odrazů se snažíme dosáhnout větší hladkosti kritériální funkce, která by nás při gradientní metodě dovedla ke správnému cíli. Uživateli je dána možnost volby nejen pro užití rozostření, ale také pro zvolení velikosti rozostření. Volí se jedno číslo určující velikost mediánového filtru, tedy velikost rozmazání.

Tlačítko s názvem ‚Spust’ výpočet‘ se aktivuje při zvolení masky. Poté tedy již stačí tlačítko spustit a čekat na výsledky optimalizace.

Všechny výsledky se zobrazují v pravé části hlavního okna, Obrázek 24. Vrchní část patří načtenému referenčnímu a registrovanému vstupnímu obrazu. Informace o jejich rozměrech nalezneme pod každým snímkem. V okně vlevo uprostřed se do grafu vykreslí ‚vymaskovaný‘ obraz vzniklý aplikací zvolené masky na referenční obrázek. Vpravo se nalézají slícované obrazy, tedy šachovnice nebo průměr referenčního obrazu a registrovaného obrazu s nejlépe ohodnocenou transformací. Spodní část patří výsledkům optimalizace. Tím jsme zjistili nejlepší dosažené hodnoty transformace a časovou náročnost průběhu algoritmu.

## **7.4 Dostupné optimalizační metody**

Pro zhodnocení správnosti a náročnosti se vytvořená funkce porovnávala s dostupnými optimalizačními metodami. Programové prostředí Matlab 2007b nabídlo vytvořené funkce `fminsearch`, `ga` a `simulannealbd`. Všechny funkce musely být upraveny pro kvalitní zhodnocení, což v sobě zahrnuje vykreslování pouze nezbytných parametrů obsažených ve všech funkcích. Jsou jimi pozice nejlepšího nalezeného bodu, jeho ohodnocení a časová náročnost metody. Každý další výpočet či vykreslení grafu podstatně zpomaluje zvolenou funkci, neboť celé výpočty probíhají v jednotkách sekund.

Všem čtyřem zpracovaným funkcím byly poskytnuty stejné podmínky v podobě připravení a vyčistění paměti, volby počátečních parametrů a vykreslení zvolené testovací funkce s konturami. Následuje zapnutí časovače, a po nastavení nezbytných parametrů, se zavolá příslušná funkce obsahující optimalizační metodu. Po vypsání výsledků v Command Windows se časovač vypne a dosažená hodnota v sekundách se zobrazí. Tento přehled je obsahem funkcí `Open_FMS`, `Open_GA` a `Open_SA`.

Do každé metody byl zanesen stochastismus v podobě náhodně zvolených počátečních bodů, aby se zabránilo ovlivňování začátku průběhu funkce. Kromě počáteční pozice se v `Fminsearch` zvolí kriteriální funkce, na kterou se metoda má aplikovat. Z nabídky možností v Genetic Algorithm and Direct Search Toolbox se zvolil takový typ simulovaného žíhání, u kterého lze navolit i hranice pro prohledávanou oblast. Příkaz optimalizační metody se nazývá `simulannealbd` a vstupními proměnnými jsou zvolená kriteriální funkce, počáteční pozice a dvě okrajové hodnoty určující rozsah prohledávaného prostoru. Kolmý rozsah pro vytvoření 2D prostoru má stejné parametry jako první, takže je zvolena čtvercová plocha.

Z důvodu porovnání našeho hybridního algoritmu (HybGA) s genetickým algoritmem (GA) programového prostředí Matlab, jsme se pro funkci `ga` snažili docílit stejného nastavení. Přes `gaoptimset` se změnila hodnota pravděpodobnosti křížení, elitářství, počet generací, velikost populace, typ selekce a migrace. Nastavení pravděpodobnosti mutace je dopočítáno automaticky z pravděpodobnosti křížení a elitářství. Opět se nastavily i hranice prohledávaného prostoru.

## 7.5 Výsledky

Vytvořený HybGA se testoval na sadě syntetických dat, do kterých jsme se snažili vložit i problémy medicínských obrazů. Vyzkoušely se také pokusy s reálnými medicínskými snímky. Středem zájmu u porovnávání s dostupnými optimalizačními technikami se stala časová náročnost výpočtu, správnost nalezení extrému a dosažené ohodnocení. Jejich výsledky jsou založené v příloze. Hlavní statistické hodnoty výsledků jsou zahrnuty v tabulkách 4-8.

Jelikož se ve všech případech jedná alespoň v malé míře o stochastismus, pro hodnocení je zapotřebí provést více měření. Základní statistické výsledky nám dají informaci o průměru, mediánu, směrodatné odchylce a rozptylu hodnot. Takto vzniklé tabulky nyní budeme analyzovat.

Odchyłka od velikosti globálního minima je ve většině případů identická s hodnotou velikosti dosaženého ohodnocení, neboť téměř všechny funkce (kromě poslední) mají globální minimum ohodnocené nulou.

Vzdálenost od skutečného extrému je důležitou veličinou, neboť slučuje dvě získané hodnoty určující polohu nalezeného bodu. Například v Tabulce 4 je u metody Fminsearch průměr výchylek bodu od extrému roven nule, což by se mohlo jevit jako velmi dobrá varianta volby. Avšak při určení vzdálenosti od skutečného extrému zjistíme, že průměrná vzdálenost je třikrát až třináctkrát větší, než je tomu u ostatních metod. Vzdálenost od skutečného extrému je vedle časové náročnosti tedy rozhodující testovací veličinou, která nabývá pouze kladných hodnot a snaží se být co nejmenší.

Rastriginova funkce má tvar strmých vlnek. Pro prohledávání jsme si zvolili prostor o velikosti  $x,y = \langle -2, 2 \rangle$ , ve kterém se nalézá 24 lokálních minim (globální minimum s hodnotou 0) a 16 lokálních maxim (globální maximum s hodnotou 4,452). Kromě dvou bodů z deseti u metody Fminsearch žádný jiný bod optimalizačních metod neuvízl v krajních falešných minimech. Funkce je pro optimalizaci náročná i v tom smyslu, že hodnoty lokálních minim jsou dosti blízké, a tak se zvětšuje pravděpodobnost uvíznutí v lokálním extrému. K nejpodobnějšímu ohodnocení jako má extrém bylo dosaženo HybGA. Naproti tomu samotný GA obsahuje nejhorší ohodnocení, neboť jeho cílem není nalezení lokálního minima.

Po stránce časové náročnosti tuto funkci snáší nejlépe Fminsearch, avšak nejméně proměnný čas je při GA. Ty jsou v počtu prohledávaných možností omezeny vstupními podmínkami o daném počtu iterací (nemusí však to být ukončovací podmínka všech genetických algoritmů [19]). Naproti tomu Fminsearch a Simulované žíhání nemají předem nastavený počet iterací, a tak jejich časová odchylka je větší.

**Tabulka 4 Rastriginova funkce; prohledává se prostor  $x,y \in \langle -2, 2 \rangle$ , extrém se nachází v  $x,y=0$  a má hodnotu 0**

	Metoda	Časová náročnost	Dosažené ohodnocení	Odchylka ohodnocení od extrému	Výchylka v ose x	Výchylka v ose y	Vzdálenost od skutečného extrému
Průměr	FS	0,199	0,179	0,179	0,000	0,000	1,259
	GA	0,843	0,190	0,190	0,428	0,224	0,570
	SA	3,647	0,050	0,050	0,198	0,099	0,440
	HybGA	0,270	0,010	0,010	0,099	0,000	0,099
Medián	FS	0,199	0,187	0,187	0,000	-0,099	1,285
	GA	0,842	0,201	0,201	0,105	0,044	0,522
	SA	3,556	0,000	0,000	0,000	0,000	0,005
	HybGA	0,274	0,000	0,000	0,000	0,000	0,000
SD	FS	0,196	0,099	0,099	0,000	-0,497	0,995
	GA	0,002	0,104	0,104	0,453	0,371	0,502
	SA	0,856	0,067	0,067	0,597	0,299	0,549
	HybGA	0,009	0,030	0,030	0,297	0,000	0,297
Rozptyl	FS	0,039	0,010	0,010	0,000	0,247	0,990
	GA	0,000	0,011	0,011	0,205	0,138	0,252
	SA	0,733	0,004	0,004	0,357	0,089	0,301
	HybGA	0,000	0,001	0,001	0,088	0,000	0,088

Podstatně odlišnou funkcí je Rosenbrockova funkce vytvářející údolí. Lokální minimum se nalézá v bodě  $x,y = \{1\}$ , kde je hodnota nulová. Minima se táhnou podél oblouku s osovou symetrií v ose x. Směrem od oblouku se hodnoty prudce zvedají a v našem zvoleném prostoru o rozměrech  $x,y \in \langle -2, 2 \rangle$  se dosahuje maxim rovnajících se 3609 jednotek.

Téměř hladký průběh funkce dává metodě Fminsearch příležitost pro výborné výsledky jak ve zvládnutí časové náročnosti, tak k nalezení globálního minima. Nejhorší hodnocení získal GA, neboť ostatní funkce se dostali alespoň do lokálního minima. HybGA téměř vždy uvízl v lokálním minimu ve vzdálenosti 0,318 jednotek od globálního.

**Tabulka 5 Rosenbrockova funkce; ; prohledává se prostor  $x,y \in [-2, 2]$ , extrém se nachází v  $x,y=1$  a má hodnotu 0**

	Metoda	Časová náročnost	Dosažené ohodnocení	Odchylka ohodnocení od extrému	Výchylka v ose x	Výchylka v ose y	Vzdálenost od skutečného extrému
Průměr	FS	0,209	0,000	0,000	1,000	1,000	0,000
	GA	0,884	0,042	0,042	0,925	0,863	0,157
	SA	3,046	0,018	0,018	0,981	0,974	0,166
	HybGA	0,269	0,021	0,021	0,859	0,736	0,299
Medián	FS	0,206	0,000	0,000	1,000	1,000	0,000
	GA	0,876	0,033	0,033	0,952	0,922	0,091
	SA	3,299	0,004	0,004	1,001	1,002	0,138
	HybGA	0,270	0,023	0,023	0,850	0,720	0,318
SD	FS	0,010	0,000	0,000	0,000	0,000	0,000
	GA	0,030	0,033	0,033	0,077	0,139	0,158
	SA	0,608	0,031	0,031	0,095	0,180	0,123
	HybGA	0,008	0,005	0,005	0,027	0,048	0,055
Rozptyl	FS	0,000	0,000	0,000	0,000	0,000	0,000
	GA	0,001	0,001	0,001	0,006	0,019	0,025
	SA	0,370	0,001	0,001	0,009	0,032	0,015
	HybGA	0,000	0,000	0,000	0,001	0,002	0,003

Mezi poslední tři testovací funkce patří Ackleyho funkce a jemu dvě podobné. Ackleyho funkci budeme opět testovat v rozměrech prostoru  $x,y \in [-2, 2]$ , kde se minimum nalézá v počátku souřadného systému a má nulovou velikost. I když funkce směrem od středu roste, obsahuje spoustu velkých lokálních minim, což znesnadňuje optimalizaci. Velikost globálního maxima závisí na zvolené oblasti. Pro námi zvolený rozsah se dosahuje největšího ohodnocení o velikosti 7,784.

**Tabulka 6 Ackleyho funkce; prohledává se prostor  $x,y \in \langle -2, 2 \rangle$ , extrém se nachází v  $x,y=0$  a má hodnotu 0**

	Metoda	Časová náročnost	Dosažené ohodnocení	Odchylna ohodnocení od extrému	Výchylka v ose x	Výchylka v ose y	Vzdálenost od skutečného extrému
Průměr	FS	0,196	4,659	4,659	0,490	0,689	1,877
	GA	0,858	1,041	1,041	0,115	0,092	0,160
	SA	2,116	3,979	3,979	-0,175	-0,140	1,113
	HybGA	0,263	0,000	0,000	0,000	0,000	0,000
Medián	FS	0,195	5,133	5,133	0,960	0,979	2,078
	GA	0,855	0,717	0,717	0,097	0,063	0,124
	SA	2,038	3,596	3,596	-0,183	-0,153	1,053
	HybGA	0,259	0,000	0,000	0,000	0,000	0,000
SD	FS	0,002	1,250	1,250	1,093	1,391	0,568
	GA	0,009	0,611	0,611	0,067	0,070	0,073
	SA	0,160	2,253	2,253	0,906	0,940	0,718
	HybGA	0,014	0,000	0,000	0,000	0,000	0,000
Rozptyl	FS	0,000	1,563	1,563	1,194	1,936	0,323
	GA	0,000	0,373	0,373	0,005	0,005	0,005
	SA	0,026	5,076	5,076	0,821	0,884	0,516
	HybGA	0,000	0,000	0,000	0,000	0,000	0,000

Obdobu Ackleyho funkce jsme podrobily většímu rozsahu  $x,y \in \langle -50, 150 \rangle$ . Jelikož poloha ani hodnota minimálního extrému se nemění, globální minimum se na obrázku přesunulo ze středu vybrané oblasti do třetího kvadrantu. Tento posun však nepotvrdil žádné větší komplikace v průběhu funkce. Zvětšení oblasti zájmu a tedy i zvětšení rozsahu mezi maximem (nyní již 309,7) a minimem (stále nulová hodnota) způsobilo zlepšení zvláště GA, neboť jeho populace obsahuje více odlišná čísla což přispívá k jasnější konvergenci do extrému. HybGA se pohoršil, neboť i když je jeho průměrné ohodnocení podobné GA, gradientní metodou sklouzne spíše nesprávným směrem. Projevuje se to větší vzdáleností od skutečného extrému.



**Tabulka 7 Ackleyho funkce upravená; prohledává se prostor  $x,y \in \langle -50, 150 \rangle$ , extrém se nachází v  $x,y=0$  a má hodnotu 0**

	Metoda	Časová náročnost	Dosažené ohodnocení	Odchyłka ohodnocení od extrému	Výchylka v ose x	Výchylka v ose y	Vzdálenost od skutečného extrému
Průměr	FS	0,200	85,094	85,094	33,276	34,039	68,704
	GA	0,854	8,630	8,630	0,100	0,090	0,153
	SA	3,117	5,416	5,416	-1,520	1,157	4,216
	HybGA	0,282	9,139	9,139	-2,100	-3,700	6,652
Medián	FS	0,198	65,771	65,771	18,956	18,956	60,723
	GA	0,850	8,980	8,980	0,087	0,057	0,157
	SA	2,969	4,288	4,288	-0,031	3,749	3,773
	HybGA	0,282	10,084	10,084	-2,000	-3,500	5,000
SD	FS	0,006	65,548	65,548	51,282	47,023	48,855
	GA	0,012	3,389	3,389	0,052	0,081	0,063
	SA	0,646	1,748	1,748	1,849	3,361	0,767
	HybGA	0,005	5,069	5,069	5,281	4,124	4,330
Rozptyl	FS	0,000	4296,559	4296,559	2629,874	2211,207	2386,785
	GA	0,000	11,485	11,485	0,003	0,007	0,004
	SA	0,417	3,057	3,057	3,418	11,297	0,588
	HybGA	0,000	25,699	25,699	27,890	17,010	18,745

Poslední testovací funkce je vytvořená tak, aby přibližně znázorňovala registraci lékařských snímků. Jedná se o nerovnou plochu s mírným poklesem ke strmému globálnímu extrému, který ovšem nemá hladký průběh. Při vytváření se vychází ze zkušenosti se snímky sítnice pořízenými fundus kamerou a z předpokladu, že lékařská data většinou nemají mírné přechody mezi objekty, takže při přesné registraci dvou snímků obvykle naskočí prudký extrém. Okrajové meze funkce jsou pro  $x,y \in \langle -100, 100 \rangle$  a globální minimum s hodnotou -0,9416 se nachází na pozici  $x,y=\{3\}$ . Většina ostatních pozic se nalézá v okolí nuly. Znovu lze poukázat na velmi dobré výsledky HybGA i samotných GA. I přes lepší ohodnocení HybGA je lepší průměrná vzdálenost nalezena u GA. Tato skutečnost je vytvořena velkou odchylkou tří hodnot. Přesvědčit se o tom můžeme z Tabulky 8, ve které je průměrná vzdálenost od extrému rovna 6 jednotkám a rozptyl je velký, avšak medián je nulový.

Fminsearch je v těchto posledních třech testovacích funkcích opět jako u Rastringa špatně využitelná, neboť podle náhodně zvolené výchozí pozice vyhledá pouze nejbližší lokální extrém, což lze číst jak z velkého dosaženého ohodnocení, velké vzdálenosti od skutečného extrému či jeho velkého rozptylu.

**Tabulka 8 Funkce podobna medicínským snímkům; prohledává se prostor  $x,y \in [-100, 100]$ , extrém se nachází v  $x,y=3$  a má hodnotu  $-0,9416$**

	Metoda	Časová náročnost	Dosažené ohodnocení	Odchylna ohodnocení od extrému	Výchylka v ose x	Výchylka v ose y	Vzdálenost od skutečného extrému
Průměr	FS	0,195	0,000	0,942	19,208	-43,528	95,352
	GA	0,903	-0,021	0,920	1,257	1,223	2,511
	SA	2,047	0,000	0,942	-24,979	-6,859	54,129
	HybGA	0,281	-0,659	0,282	7,300	-0,700	6,391
Medián	FS	0,192	0,000	0,942	24,043	-59,910	92,678
	GA	0,848	0,000	0,942	1,115	1,111	2,642
	SA	1,983	0,000	0,942	-4,490	-10,424	52,007
	HybGA	0,281	-0,942	0,000	3,000	3,000	0,000
SD	FS	0,005	0,000	0,000	60,947	56,006	19,405
	GA	0,164	0,064	0,064	0,586	0,373	0,609
	SA	0,189	0,000	0,000	39,573	44,851	39,086
	HybGA	0,007	0,431	0,431	8,198	7,484	10,703
Rozptyl	FS	0,000	0,000	0,000	3714,536	3136,688	376,551
	GA	0,027	0,004	0,004	0,343	0,139	0,371
	SA	0,036	0,000	0,000	1566,006	2011,586	1527,730
	HybGA	0,000	0,186	0,186	67,210	56,010	114,558

## Testování algoritmu na obrazech

Kontrola kvality vytvořeného HybGA se provedla i na obrazech. Testování se provádělo přes vzdálené připojení na dvouprocesorový stroj s 64GB RAM. K dispozici byly dva vytvořené testovací obrazce, kterým se věnovala kapitola 6.2. Jelikož se optimalizace má vztahovat na registraci medicínských snímků, kvalita algoritmu HybGA se vyzkoušela i na snímku sítnice vytvořeného fundus kamerou. Tento snímek se převzal z bakalářské práce [30]. Další obrazová data se získala během průběhu praxe ve výzkumné divizi firmy Biovendor. Jedná se o snímky pořízené fluorescenčním mikroskopem na kterých je znázorněna experimentální buněčná linie 293 HEK (Human Embryonal Kidney). Svůj kulatý tvar si zachovávají díky rotačním nádobám. Při ponechání v klidu totiž buňky přisedají a jejich okrajové části se začínají protahovat do filopodií.

Při testování se pro každé zvolené nastavení parametrů provádělo deset měření. Jejich hodnoty jsou zapsány v souboru Výsledky na přiloženém CD. Z důvodu velkého množství dat se zde uvádějí jen některá statistická data. Cílem bylo zjistit přibližnou souvislost mezi správnou funkčností algoritmu a volbou některých parametrů. Hlavní zaměření bylo na rozostření, velikost počáteční populace a počet iterací.

Kvalita nastavení parametrů je popsána na základě vzdálenosti transformace dvou obrazů od jejich ideálního slícování, a to jak pro celý algoritmus HybGA (Vzdálenost HybGA), tak pro jeho část bez gradientní metody (GA). Tím jsme se chtěli ujistit, zda použitím hybridní metody opravdu dochází ke zlepšení optimalizace. O velikosti zlepšení či zhoršení pojednává Rozdíl vzdáleností. Nevychází však z rozdílu průměrných vzdáleností, ale z rozdílu jednotlivých vzdáleností, z kterých vytvoří průměrnou hodnotu. To je důvodem, proč například v Tabulce 10 se čtvrtý sloupec nemůže vypočítat z předchozích dvou. Pro představu se uvádí statistické zpracování i velikosti ohodnocení (Ohodnocení), časové náročnosti (Čas) a počtu kroků gradientní metody k nalezení extrému (Počet dalších výpočtů). Čas je závislý nejen na počtu dalších výpočtů, ale také na velikosti populace a počtu iterací.

Velikost rotace se do parametru vzdálenosti od ‚ideální‘ pozice nezapočítává, neboť na ni díky středové osy otočení buď nemá vliv nebo její vliv je pouze minimální (v případě úhlopříček).

Tabulka 10 je legendou pro Tabulku 9. Pod proměnnou NP (nastavení parametrů) se nachází zkratky šesti nastavení. T značí použití testovacího obrazce s tečkami a P označuje testovací obrazec s pruhy. Dané masky lze nalézt na příloženém CD ve složce Zdrojové soubory.

**Tabulka 9 Statistické ohodnocení algoritmu HybGa s volbou různých parametrů při použití testovacích obrazců**

	NP	Vzdálenost HybGA	Vzdálenost GA	Rozdíl vzdáleností	Ohodnocení	Čas	Počet dalších výpočtů
Průměr	T1	6,473	11,531	6,316	1,885	11,676	7,300
	T2	13,482	13,059	2,264	1,692	19,178r	4,900
	T3	14,640	14,900	3,886	1,761	8,676	6,100
	P1	6,536	7,161	1,258	1,679	317,299	4,100
	P2	10,576	10,800	2,485	1,866	46,258	5,300
	P3	5,188	6,752	2,050	1,785	392,339	4,900
Medián	T1	0,000	10,625	3,606	2,000	11,214	6,500
	T2	13,000	14,123	1,569	1,664	19,100	4,500
	T3	13,000	14,018	4,932	1,666	8,446	7,500
	P1	6,000	5,957	1,152	1,669	292,948	4,000
	P2	11,148	11,073	1,846	1,845	44,347	5,000
	P3	6,000	6,155	1,214	1,667	368,458	5,000
SD	T1	8,701	7,074	6,790	0,176	3,317	2,830
	T2	5,093	4,535	3,205	0,103	3,507	1,700
	T3	10,018	10,009	2,484	0,157	1,777	2,385
	P1	4,896	4,292	1,060	0,114	81,780	0,943
	P2	6,690	4,937	1,983	0,078	12,720	1,952
	P3	4,763	3,566	2,098	0,178	109,131	1,640

**Tabulka 10 Legenda pro nastavení parametrů (NP) při použití testovacích obrazců (T-tečky, P-pruhy) vzorkování je vždy 100%**

NP	Velikost populace	Počet iterací	Maximální záporný chromosom	Maximální kladný chromosom	Maska	Rozostření
T1	30	12	[-30, -30, 0]	[30, 30, 0]	maska_tecky1.mat	Obojí 3
T2	30	12	[-30, -30, 0]	[30, 30, 0]	maska_tecky1.mat	Žádné
T3	30	4	[-30, -30, 0]	[30, 30, 0]	maska_tecky1.mat	Žádné
P1	15	6	[-15, -15, -2]	[15, 15, 2]	Maska_pruhy1.mat	Oboje 9
P2	30	12	[-30, -30, 0]	[30, 30, 0]	Maska_pruhy1.mat	Žádné
P3	15	6	[-15, -15, -2]	[15, 15, 2]	Maska_pruhy1.mat	Žádné

Z Tabulky 10 lze vyčíst, že nejlepších výsledků se dosáhlo při použití obrazců s tečkami při parametrech T1. Uvádí se maximální ohodnocení průměrné s mediánem rovným hodnotě 2. Z velkého rozdílu vzdáleností a množství dalších výpočtů lze vyčíst, že tak vysokého ohodnocení se dosáhlo zejména díky gradientní metodě. Na velmi dobré výsledky ohodnocení poukazuje i volba P2, avšak se pravděpodobně bude jednat o uvíznutí v lokálním minimu, neboť vzdálenost od globálního extrému je velká, jak ve vyjádření průměrem, tak mediánem. Z tabulek obsažených na CD je vidět, že i při téměř maximálních hodnotách chromosomu se dosáhlo globálního extrému. Je to důsledek lícování pravidelného testovacího obrazce, jehož kritériální funkce neobsahuje pouze jeden extrém. Při dobře nastavené velikosti masky či výběru menší oblasti pro transformaci se lze této chyby vyvarovat. Velká časová náročnost u P1 a P3 je způsobena umožněním rotace od úhlu  $-2^\circ$  do  $2^\circ$ .

Při použití HybGA na identické snímky sítnice se zaměřilo hlavně na volbu rozostření a pozice masky. První maska je vytvořena z oblasti disku zrakového nervu (maaa.mat) a druhá obsahuje rozbíhání velké cévy (cevy.mat). Z důvodu velkého množství dat se vytvořila i Tabulka 13, která zahrnuje všechny výsledky působení algoritmu na snímek sítnice bez ohledu na volbu parametrů nastavení.

**Tabulka 11 Legenda pro nastavení parametrů (NP) při použití snímku sítnice; vzorkování je vždy 20%**

NP	Velikost populace	Počet iterací	Maximální záporný chromosom	Maximální kladný chromosom	Maska	Rozostření
1	10	4	[-30, -30, 0]	[30, 30, 0]	maaa.mat	Obojí 6
2	10	4	[-30, -30, 0]	[30, 30, 0]	maaa.mat	Obojí 9
3	10	4	[-30, -30, 0]	[30, 30, 0]	maaa.mat	Obojí 3
4	10	4	[-30, -30, 0]	[30, 30, 0]	maaa.mat	Žádné
5	30	4	[-30, -30, 0]	[30, 30, 0]	maaa.mat	Žádné
6	10	12	[-30, -30, 0]	[30, 30, 0]	maaa.mat	Obojí 6

7	30	12	[-30, -30, 0]	[30, 30, 0]	maaa.mat	Obojí 6
8	10	4	[-30, -30, 0]	[30, 30, 0]	cevy.mat	Žádné
9	10	4	[-30, -30, 0]	[30, 30, 0]	cevy.mat	Obojí 6

**Tabulka 12** Statistické ohodnocení algoritmu HybGa s volbou různých parametrů při optimalizaci snímku sítnice

	NP	Vzdálenost HybGA	Vzdálenost GA	Vzdálenost HybGA a GA	Ohodnocení	Čas	Počet dalších výpočtů
Průměr	1	6,762	12,808	6,046	1,811	31,677	7,200
	2	4,755	8,721	3,966	1,812	29,463	5,300
	3	8,784	13,061	4,347	1,807	29,707	6,100
	4	5,890	11,499	5,609	1,808	34,427	6,300
	5	2,864	8,482	5,618	1,936	81,584	6,500
	6	7,759	13,136	5,378	1,745	61,266	6,900
	7	3,347	9,082	5,735	1,873	201,273	6,600
	8	2,385	6,680	4,504	1,935	31,184	5,800
	9	5,662	8,281	4,290	1,871	30,028	5,500
Medián	1	0,000	9,610	7,073	2,000	31,420	8,000
	2	0,000	5,712	3,581	2,000	28,832	5,500
	3	0,000	9,137	2,921	2,000	28,650	4,500
	4	0,000	10,374	3,864	2,000	35,815	4,500
	5	0,000	6,698	5,712	2,000	75,555	7,000
	6	0,000	12,405	4,337	2,000	63,525	6,000
	7	0,000	9,192	5,338	2,000	192,770	7,000
	8	0,000	3,864	3,217	2,000	29,675	4,000
	9	0,000	4,718	2,921	2,000	27,045	4,500
SD	1	10,903	8,698	3,721	0,289	10,000	3,400
	2	8,491	7,413	3,076	0,287	6,547	2,410
	3	13,596	10,996	4,193	0,294	8,673	4,230
	4	9,269	7,541	4,787	0,294	10,797	4,196
	5	8,591	7,580	3,979	0,193	20,796	3,202
	6	10,460	7,832	4,667	0,313	11,510	4,888
	7	6,745	5,350	4,054	0,254	20,583	2,835
	8	7,156	6,639	4,065	0,194	10,235	3,250
	9	11,421	9,307	3,523	0,257	11,048	2,907

**Tabulka 13** Statistické ohodnocení algoritmu HybGa při optimalizaci snímku sítnice bez ohledu na zvolené parametry

	Vzdálenost HybGA	Vzdálenost GA	Rozdíl vzdáleností	Ohodnocení	Čas	Počet dalších výpočtů
Průměr	5,356	10,194	5,055	1,844	58,957	6,244
Medián	0,000	7,968	4,329	2,000	57,032	5,667
SD	9,626	7,929	4,007	0,264	12,243	3,480

Při použití HybGA na identické snímky sítnice lze vidět velmi dobré výsledky při všech nastaveních. Pojednává o tom vysoké ohodnocení a malá hodnota vzdálenosti HybGA.

Mezi nejlepší nastavení se řadí ND8 a ND5. Lze mluvit o lepších výsledcích při použití gradientní metody než v případě samotného genetického algoritmu.

Posledními testovacími daty byla dvojice snímků buněk. Odlišnost je většinou způsobena změnou zaostření, osvětlení nebo pohybem buněk či nečistot v Bürkerově komůrce.

**Tabulka 14 Legenda pro nastavení parametrů (NP) při použití snímku buněk; vzorkování je vždy 50%**

NP	Velikost populace	Počet iterací	Maximální záporný chromosom	Maximální kladný chromosom	Maska	Rozostření
1	30	6	[-30, -30, 0]	[30, 30, 0]	maska_bunky3.mat	Obojí 3
2	30	12	[-30, -30, 0]	[30, 30, 0]	maska_bunky3.mat	Obojí 3
3	30	12	[-30, -30, 0]	[30, 30, 0]	maska_bunky3.mat	Žádné
4*	30	12	[-30, -30, 0]	[30, 30, 0]	maska_bunky3.mat	Žádné

**Tabulka 15 Statistické ohodnocení algoritmu HybGA s volbou různých parametrů při optimalizaci dvou mírně odlišných snímků buněk**

	NP	Vzdálenost HybGA	Vzdálenost GA	Vzdálenost HybGA a GA	Ohodnocení	Čas	Počet dalších výpočtů
Průměr	1	9,456	8,851	1,403	1,441	33,048	4,200
	2	8,155	8,894	1,263	1,446	53,219	4,300
	3	5,429	7,648	2,398	1,468	56,235	4,700
	4*	8,712	8,581	1,766	1,458	71,021	4,100
Medián	1	10,018	8,911	0,999	1,422	30,534	4,000
	2	8,000	8,154	0,804	1,423	50,432	4,000
	3	0,000	4,352	2,236	1,498	55,303	3,500
	4*	6,023	5,338	1,207	1,449	65,539	4,000
SD	1	6,090	4,853	1,004	0,039	10,275	1,326
	2	6,099	4,939	1,207	0,038	9,248	1,268
	3	7,837	6,856	1,950	0,036	7,971	2,238
	4*	9,788	8,767	1,420	0,034	13,198	1,374

Hvězdička u čtvrtého typu nastavení parametrů signalizuje použití zvláštní části algoritmu, ve které se během každé iterace vytvoří průměr dvou nejlepších ohodnocení. Při zjištění případného nového extrému se opět provede elitismus. Jelikož nebylo potvrzené výrazné zlepšení optimalizace, zmínka o jeho existenci se nachází pouze zde. Z tabulky je patrné, že svou účastí i prodlužuje časovou náročnost.

## 8 Diskuse

Z výsledků testování nevzešla pro univerzální použití žádná metoda. Vždy záleží na požadavcích uživatele. Zda potřebuje nalézt alespoň podobnou optimalizaci (příliš mu nevádí nalezení lokálního minima), co nejpřesnější optimalizaci (potřebuje se dostat co nejbližší globálnímu extrému), nebo zda upřednostňuje malou časovou náročnost. V případě znalosti tvaru kriteriální funkce nebo odhadnutého tvaru z registrace snímků je volba správné optimalizační metody snadnější.

U metod s ukončující podmínkou typu dosažení nejnižší hodnoty, může snadno dojít k uvíznutí v lokálním extrému, který může být více vzdálen od skutečného. Tento problém nikdy nenastal pouze u GA, který je ukončen dosažením předem zvoleného počtu generací. Využití této vlastnosti přinášelo dobré výsledky zvláště u funkcí typu Ackley, kdy se funkce svažuje do globálního minima, ale přitom obsahuje spoustu menších lokálních minim. Nikdy se však nastaveným počtem generací na hodnotu pět a velikosti počáteční populace o padesáti jedincích nedosáhlo extrému. Lepší výsledky v tomto ohledu přinášel námi vytvořený HybGA, který díky přidané gradientní metodě se ukončil nalezením minima, avšak ne vždy to bylo minimum globální, čímž se jeho pozici mohl vzdálit. Je to zřejmé z tabulek výsledků, ve kterých HybGA má nižší průměrnou hodnotu minima, avšak průměrná vzdálenost od pozice skutečného extrému je větší. V opačném případě HybGA vykazuje mnohem lepší výsledky než GA u funkcí s malými výchylkami ohodnocení poblíž globálního extrému, neboť ty dokáže překonat.

Velké rozdíly mezi výsledky všech metod při použití podobných funkcí Rastrigina a Ackleyho lze vysvětlit následovně. Pravidelné extrémy v Rastriginově funkci mají téměř srovnatelnou hodnotu, což zapříčiní zmatení vyhledávání u všech metod optimalizace. Přitom každé minimum je obklopeno prudce vzrůstajícím maximem, takže i malá změna posunu může způsobit velkou změnu ohodnocení. Tyto dva fakty způsobí, že průběh metod bude vykazovat lepší výsledky u Ackleyho funkce. I přes zmíněné problémy měla nejlepší výsledky v Rastriginove funkci HybGA, jejíž průběh dokázal nalézt správnou oblast a dostat se až do její minimální hodnoty.

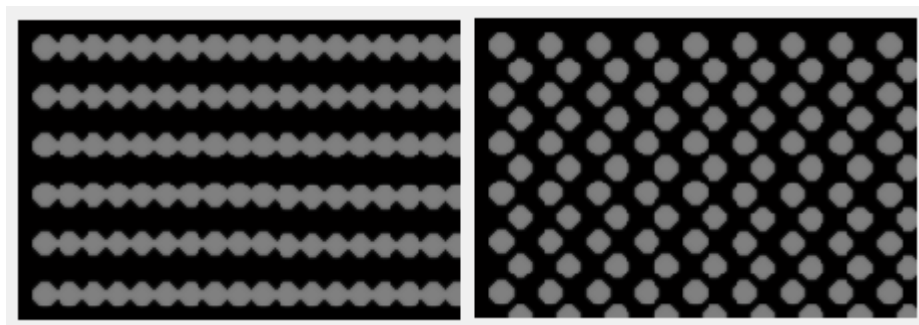
Výsledky metody Fminsearch odpovídají jejímu charakteru optimalizace, kdy v prostoru funkce postupuje gradientem z počáteční pozice do místa minima. Úspěšná může být pouze tehdy, pokud je funkce hladká a obsahuje pouze jedno minimum. Takovou je První De Jungova funkce, kterou jsme kvůli její jednoznačnosti netestovali, a případně i Rosebrockova funkce, u které však mohl nastat problém mírných výchylek v oblouku s malým ohodnocením. Jediným parametrem, ve kterém funkce vždy vynikala nad ostatní, byla malá časová náročnost.

Nejhorší časovou náročností a její směrodatnou odchylkou disponuje metoda simulovaného žíhání, jejíž průměrný čas byl desetkrát až patnáctkrát delší než průběh

nejrychlejší metody. Po stránce nalezení globálního minima nikdy nezaujala první místo, ale nikdy se nestala ani nejhorší metodou. Nejvíce jí vyhovují funkce s výraznějším rozsahem hodnot mezi globálním maximem a minimem, jakou je upravená Ackleyho funkce s větším prohledávaným prostorem.

Časová náročnost u GA a HybGA je v porovnání s metodou simulovaného žíhání velmi dobrá. V závislosti na zvolené kritériální funkci je zlepšení u GA trojnásobné a u HybGA desetinásobné. Důvodem je fakt, že již v počáteční populaci se získají informace o přibližném rozložení funkční hodnoty, a tudíž lze rychleji konvergovat ke správnému řešení. Výrazná změna času při použití různých kritériálních hodnot nenastává, neboť genetické algoritmy jsou svou podstatou založené na předem stanoveném počtu velikosti populace a počtu generací. Případná malá změna může nastat při rozdílném počtu jedinců vybraných ke křížení či mutaci, neboť tyto hodnoty nejsou vždy stejné, ale jsou závislé na pravděpodobnostních parametrech.

Výsledky testovacích obrazců poukazují na jejich dosti složitou optimalizaci, neboť jejich pravidelnost bez velké tvarové rozmanitosti dává vzniknout velmi podobným extrémům. V případě použití vzájemné informace pro registraci se může mluvit o falešných extrémech, neboť vzájemné histogramy nerozlišují odstín, ale počet pixelů daného odstínu. Tento případ se vyskytoval zejména u testovacího obrazce s tečkami, kdy velké množství optimalizací skončilo překrytím tečky ve středu černé oblasti, jako je tomu na Obrázku 25. Zobrazené slícování je vytvořené průměrnými hodnotami pixelů obou obrázků, proto jsou vidět všechny objekty, avšak hodnota bílých bodů klesla na 128.



**Obrázek 25 Registrace testovacích obrazů na falešných extrémech; vlevo transformace plovoucího obrazu o  $x=13, y=0$ ; vpravo transformace obrazu o  $x=13, y=-10$**

Výsledky jsme se přesvědčili, že je velmi důležité uvážlivě volit masku a velikost prohledávaného okolí, aby registrací nevznikl extrém podobný globálnímu. To by mohlo nastat v důsledku většího množství velmi podobného okolí, jako je obsah masky.

V tomto případě testovacích obrazců s velkými pravidelnými objekty, které mají ostré ohraničení lze mluvit o lepších výsledcích díky rozostření obou obrazů. Vytvoří se tím totiž hladší kritériální funkce, podle které může HybGA lépe postupovat k extrému.



Vytvořený HybGA se velmi dobře osvědčil na medicínském snímku znázorňující sítnici. Ideální slícování dvou identických snímků nastalo v nadpoloviční většině (až 90%), což se projevilo mediánem ohodnocení o velikosti 2. Vyšší číslo ohodnocení se při použití vzájemné informace nevyskytuje. Dokonalému slícování nasvědčuje i fakt nulového mediánu vzdálenosti od extrému, což se vypočítalo u všech případech nastavení parametrů.

Porovnáním výsledků čtvrtého až sedmého řádku jsou zjištěny lepší výsledky při volbě větší počáteční populace. Při menším počtu populací může dojít k nedokonalému průzkumu oblasti, čímž může nastat uvíznutí v lokálním extrému. O velikosti populace pojednává i směrodatná odchylka, jejíž hodnota se u větších populace zmenšuje. Nemusí však platit, že malá populace bude mít vždy vysokou směrodatnou odchylku. Na získaných datech Tabulky 12 lze odhadovat i velkou výchylku několika chromosomů, které zbytečně zvětšily průměrnou vzdálenost od ‚ideální‘ pozice optimalizace.

Při zkoumání vztahu mezi HybGA a volbou rozostření podstatná změna nenastala. Mírně lepší výsledky se nalézají u původního ostrého snímku. Může to být způsobeno malými detaily, které nezpůsobily příliš velkou změnu ohodnocení.

Nejlepší výsledky optimalizace nastávají na ostrých snímkách, kdy se volí buď vyšší velikost počáteční populace nebo maska je zaměřená na oblast s výraznějšími prvky-cívami. Při trojnásobném zvětšení populace nastalo přibližně trojnásobné prodloužení výpočetního času.

Tabulkou potvrzená je i logická úvaha o přímé úměře mezi počtem dalších výpočtů a vzdáleností extrémů nalezených pomocí GA od HybGA.

Z dat optimalizace buněk je špatná registrace jasně viditelná. Maximální ohodnocení v tomto případě není o velikosti 2 ale může dosáhnout pouze hodnoty 1,499. Je to způsobeno nenulovou diferencí registrace v oblasti masky. Příkladem je Obrázek 26, na kterém v levé části vidíme světlejší vymaskovaný snímek s odlišnou pozicí jedné buňky, než je na snímku vpravo. Z důvodu malých objektů jsou lepší výsledky optimalizace na ostrých snímcích.



**Obrázek 26** Invertované vymaskované snímky buněčné linie s odlišnou pozicí jedné buňky

## Závěr

V diplomové práci jsme se zabývali optimalizací postavenou na genetických algoritmech. Nejprve se připomněly základy registrace obrazů, do které spadají pojmy transformace, interpolace a kriteriální funkce. Ty se již objevily v bakalářské práci, která se zabývala hodnocením kriteriálních funkcí pro registraci obrazů sítnice. Hlavní problematikou registrace digitálních obrazů je vzorkování a kvantování, při nichž dochází ke ztrátě či zkreslení informací. Časová náročnost je dalším problémem registrace. Přímou úměrou ji ovlivňuje nejen počet pixelů obrazu, ale také počet parametrů transformace. V případě rotace a nonrigidních transformací se většinou musí užít i interpolace, což představuje další časové prodloužení výpočtu.

Pojem optimalizace v sobě skrývá algoritmické postupy, které se snaží co nejefektivněji řešit daný problém. Dvěma protipóly v těchto situacích bývají správnost nalezení globálního extrému a časová náročnost. Při prohledávání hrubou silou se vždy dojde ke správnému výsledku, avšak v objemných datech není tato metoda prakticky využitelná. Proto byla vymyšlena spousta jiných metod, které buď náhodně nebo řízeně prohledávají prostor, a při naplnění ukončovací podmínky podají informace o nejlepším nalezeném řešení. Lze mluvit o deterministických metodách, které při stejných počátečních podmínkách vyhledají stejné řešení, či o stochastických metodách, obsahujících prvek náhodnosti. Proto je jejich výsledek i při stejných počátečních podmínkách odlišný. Mají však vyšší tendenci prohledání oblasti a zabránění uvíznutí ve stejném lokálním extrému.

Další kapitola se zabývala metodou genetických algoritmů. Teoretický základ metody čtenáře seznámil s pojmy genetických algoritmů, které názorně vysvětluje na příkladech. Jelikož se kromě ohodnocení jedinců nevyužívá žádné apriorní informace, genetické algoritmy jsou ideálními nástroji v případech, kdy žádná technika řešení není k dispozici nebo není ani známa. Specializované algoritmy pro danou situaci však ve své kvalitě nepředčí. Proto je oblíbeným tématem hybridizace genetických algoritmů, která úspěšně spojuje několik metod nebo jejich částí dohromady za účelem vytvoření lepšího algoritmu.

V navazující kapitole se zmínily čtyři hlavní oblasti užití globálních optimalizačních metod při registraci medicínských obrazů. Jejich hlavním účelem je napomáhání doktorům v rozhodování formou poskytnutí ucelených informací o nálezech na snímcích.

V rámci teoretické části se provedl návrh genetického algoritmu. Byly vytvořeny i testovací funkce a testovací obrazce pro zkoušení funkčnosti a správnosti optimalizace. Dbalo se na extrémní případy registrace medicínských dat. Testovací obrazce byly podrobeny vzájemné informaci k dosažení dvourozměrných funkcí znázorňujících lícování.

Praktická část spočívala v realizaci navrženého genetického algoritmu v programovém prostředí Matlab a jeho porovnání s jinými metodami.

Vytvořený program se skládá z načítacího skriptu a do něho vložené funkce se složitou vnitřní strukturou. V načítacím skriptu jsou obsaženy testovací funkce a proměnné s volbou počátečních parametrů. Hlavní funkce po načtení vstupních parametrů vytvořila počáteční populaci, která se během cyklu selekce, křížení a mutace zkvalitňovala. Typem selekce byl zvolen souboj, neboť se u něj prokázala malá časová náročnost. Po každém zásahu do populace se ověřilo uložení nejsilnějšího jedince, který na konci každé iterace nahradil jedince s nejnižším ohodnocením. Průběh velikosti kvality populace byl znázorněn grafem.

Vytvořená metoda se zpřesnila přidáním iterační gradientní metody, která prohledává své okolí a postupuje po gradientu do extrémní hodnoty. Dalším ověřeným zlepšením se stalo ignorování jedné hodnoty nevyhovující gradientu. Tím bylo umožněno překonat drobný lokální extrém a opět postupovat po směru gradientu. V případě navrácení se do již prohledané polohy, metoda skončí a vypíše se výsledek. Pro názornost bylo vytvořeno okno, ve kterém lze vidět průběh i výsledek optimalizace.

Kvalita navrženého algoritmu se porovnávala s metodami dostupnými z Genetic Algorithm and Direct Search Toolbox v Matlabu. Jednalo se o simulované žihání s nastavenými hranicemi prohledávaného prostoru (simulannealbnd), metodu nalezení lokálního extrému (fminsearch) a genetický algoritmus (ga), jehož parametry se pomocí gaoptimset nastavily na hodnoty ekvivalentní s námi vytvořeným algoritmem. Pro kvalitní zhodnocení se dbalo na stejné požadavky pro všechny metody. Každá metoda má tedy svůj otvírací skript s funkcemi a počátečními hodnotami. Výsledky poukazují na kvalitní optimalizaci správným výběrem funkce. Tudíž nelze mluvit o nejlepší metodě, ale o použitelnosti metody na danou funkci. Pro hladké funkce s jedním globálním minimem je malé časové náročnosti ideální fminsearch. Metoda simulovaného žihání dosahuje podobných optimalizačních výsledků jako genetické algoritmy, ale ve funkcích s neměnnou hodnotou se hrouť. Z hlediska časové náročnosti se stala nejhorší metodou. Genetický algoritmus se vždy přiblížil extrému, ale při daných parametrech jej mohl dosáhnout pouze gradientní metodou. Časová náročnost byla nízká a téměř neměnná. Námi vytvořená metoda i přes přidanou gradientní metodu vykazovala k nalezení optima čtyřikrát kratší čas, než bylo potřeba k získání výsledku metodou genetického algoritmu.

Pro práci s obrazy se vytvořilo grafické uživatelské prostředí s možností nastavení parametrů podle aktuálních potřeb uživatele. Pro tento účel se zcela přetvořil načítací skript a v hlavním skriptu přibyla vložená funkce Registruj pro transformaci a ohodnocení chromosomů. Výstupem se získaly informace o hodnotách nejlepší nalezené transformace a data, která názorně vytvoří výsledek optimalizace pomocí šachovnice nebo průměru.

Testování hybridního genetického algoritmu pomocí obrazů se provádělo připojením na vzdálenou plochu dvouprocesorového přístroje. Celkový čas průběhu a kvalita optimalizace závisely na řadě parametrů. My jsme se při testování zaměřili zvláště na vztah rozostření k výsledku optimalizace. Lepší výsledky byly zjištěny při použití ostrých snímků, neboť se nepotlačily malé detaily. Avšak u snímků s pravidelnými, velkými a dobře ohraničenými objekty bylo zapotřebí menšího rozostření. Maska se musí volit s ohledem na neměnnou informaci v obou obrazech, jinak se hodnota globálního extrému dosti sníží, a bude problém ji odlišit od lokálních extrémů. Z podobného důvodu nesmí obsahovat stejná nebo dosti podobná data, jaké jsou v prohledávané oblasti. Buď se musí vybrat jiná část pro vytvoření masky, nebo se musí oblast zmenšit. Nejlepších výsledků se dosáhlo při optimalizaci dvou identických snímků sítnice, kdy se maskou zvolila zajímavá oblast s cévami rozcházejícími se do různých stran, s nulovým rozostřením a o větším počtu populace. V tomto případě se dosahovalo kvalitní optimalizace v 90%.

Vytvořený algoritmus tedy obhájil svoji konkurenceschopnost při porovnávání s jinými metodami.

## Seznam literatury

- [1] AIT-AOUDIA, S, MAHIOU, R. Medical image registration by simulated annealing and genetic algorithms. Algeria. 2007. Dostupné z WWW: <[ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4271735](http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4271735)>
- [2] AIT-AOUDIA, S, MAHIOU, R.; , "Evaluation of medical image registration by some meta-heuristics," Communications, Control and Signal Processing, 2008. ISCCSP 2008. 3rd International Symposium on , vol., no., pp.107-110, 12-14 March 2008 doi: 10.1109/ISCCSP.2008.4537201 Dostupné z WWW: <[ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4537201&isnumber=4537177](http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4537201&isnumber=4537177)>
- [3] ANRONG YANG; LINGQI MENG; JIANZHEN LUO; CAIXING LIN; , "A Rapid Registration Framework for Medical Images," Image and Graphics, 2007. ICIG 2007. Fourth International Conference on , vol., no., pp.731-736, 22-24 Aug. 2007 doi: 10.1109/ICIG.2007.78 URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4297177&isnumber=4297037>
- [4] AUNG, M.S.H.; GOULERMAS, J.Y.; HAMDY, S.; POWER, M.; , "Spatiotemporal Visualizations for the Measurement of Oropharyngeal Transit Time From Videofluoroscopy," Biomedical Engineering, IEEE Transactions on , vol.57, no.2, pp.432-441, Feb. 2010 doi: 10.1109/TBME.2009.2019828 URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4814700&isnumber=5393902>
- [5] BARTOŠ, Michal. Lícování obrazů lékařských zobrazovacích modalit. [s.l.], 2007. 66 s. Diplomová práce.
- [6] BAUM, K.G.; RAFFERTY, K.; HALGUERA, M.; SCHMIDT, E.; , "Investigation of PET/MRI image fusion schemes for enhanced breast cancer diagnosis," Nuclear Science Symposium Conference Record, 2007. NSS '07. IEEE , vol.5, no., pp.3774-3780, Oct. 26 2007-Nov. 3 2007 doi: 10.1109/NSSMIC.2007.4436944 URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4436944&isnumber=4436808>
- [7] DAŇĚK, Ondřej. Registrace obrazů buněk podle intenzit v obraze. [s.l.], 2006. 63 s. Diplomová práce. Dostupný z WWW: <[is.muni.cz/th/60931/fi\\_m/xdanek2-dp.pdf](http://is.muni.cz/th/60931/fi_m/xdanek2-dp.pdf)>
- [8] DLUHOŠ, Petr. Metaheuristické optimalizační metody pro registraci obrazů z magnetické rezonance. 2011. Bakalářská práce. Dostupné z WWW: <[http://is.muni.cz/th/269281/prif\\_b\\_b1/bakalarskaPrace.pdf](http://is.muni.cz/th/269281/prif_b_b1/bakalarskaPrace.pdf)>
- [9] GOLDBERG, David, DEB, Kalyanmoy. A comparative analysis of selection schemes used in genetic algorithm. Foundations of genetic algorithm. San Mateo : Morgan Kaufmann Publisher, Inc., 1991. ISBN 1-55860-170-8. Dostupný z WWW: <[citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.101.9494](http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.101.9494)>
- [10] HYNEK, Josef. Genetické algoritmy a genetické programování. Praha : Grada Publishing, a.s., 2008. 200 s. ISBN 978-80-247-2695-3.
- [11] HUANG, Chung-Hsien, LEE, Jiann-Der, Juany, HUANG Jau-Hua. Registration of CT image and facial surface data used adaptive genetic algorithm. Tao-Yuan : 2005 ] [cit. 2009-11-14]. Dostupný z WWW: [www.worldscinet.com/bme/17/1704/open-access/S1016237205000317.pdf](http://www.worldscinet.com/bme/17/1704/open-access/S1016237205000317.pdf)
- [12] CHEN, Y-W, MIMORI, A. Hybrid particle swarm optimization for medical image registration. Fifth International Conference On Natural Computation. Japan. 2009. Dostupný z WWW: <[ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5363336](http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5363336)>

- [13] CHEN, Yang, QIN, Zheng, SONG, Mingyue. A Continuous Medical Image Registration Approach Based on Image Segmentation and Genetic Algorithm, Information Engineering and Computer Science, 2009. ICIECS 2009. International Conference on, vol., no., pp.1-4, 19-20 Dec. 2009 doi: 10.1109/ICIECS.2009.5363347 Dostupný z WWW: [ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5363347&isnumber=5362514](http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5363347&isnumber=5362514)
- [14] INGOLE, V.T.; DESHMUKH, C.N.; JOSHI, A.; SHETE, D.; , „Medical Image Registration Using Genetic Algorithm,“ Emerging Trends in Engineering and Technology (ICETET), 2009 2nd International Conference on , vol., no., pp.63-66, 16-18 Dec. 2009 doi: 10.1109/ICETET.2009.143 URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5395423&isnumber=5394994>
- [15] JAN. Jiří. Číslíková filtrace, analýza a restaurace signálů. Brno : VUTIUUM, 2002. ISBN 80-214-2911-9
- [16] JUREK, M. Numerické metody optimalizace. 2007. Diplomová práce. Dostupné z WWW: < [dspace.knihovna.utb.cz/bitstream/handle/10563/3990/jurek\\_2007\\_dp.pdf?sequence=1](http://dspace.knihovna.utb.cz/bitstream/handle/10563/3990/jurek_2007_dp.pdf?sequence=1) >
- [17] KARKAVITSAS, G, RANGOUSI, M. Object localization in medical images using genetic algorithms. 2005. Dostupný z WWW: < [www.waset.org/journals/waset/v2/v2-2.pdf](http://www.waset.org/journals/waset/v2/v2-2.pdf) >
- [18] KOZUMPLÍK, J. Evoluční algoritmy 2. Přednáška. [cit. 2011-11-20]
- [19] KOZUMPLÍK, J. Evoluční algoritmy 3. Přednáška. [cit. 2011-11-20]
- [20] KUBEČKA, Libor. OPTIMALIZATION METHODS FOR IMAGE REGISTRATION. In Sborník prací konference a soutěže Student EEICT 2003. [s.l.] : [s.n.], 2003. s. 24. ] [cit. 2009-12-12]. Dostupný z WWW: <[www.feec.vutbr.cz/EEICT/2003/fsbornik/02.../05-kubecka\\_libor.pdf](http://www.feec.vutbr.cz/EEICT/2003/fsbornik/02.../05-kubecka_libor.pdf)>.
- [21] KVASNIČKA, V, POSPÍCHAL, J, TIŇO P. Evolučné algoritmy. Bratislava : Tlač Vydavateľstvo STU, 2000. ISBN 80-227-1377-5
- [22] LI NAN; GAO PENG DONG; LU YONGQUAN; YU WENHUA; , "The Implementation and Comparison of Two Kinds of Parallel Genetic Algorithm Using Matlab," Distributed Computing and Applications to Business Engineering and Science (DCABES), 2010 Ninth International Symposium on , vol., no., pp.13-17, 10-12 Aug. 2010 doi: 10.1109/DCABES.2010.9 URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5572154&isnumber=5570815>
- [23] MANSOORI, T.; PLANK G; BURTON, R.; SCHNEIDER, J.; KOHL, P.; GAVAGHAN, D.; GRAU, V.; , "AN ITERATIVE METHOD FOR REGISTRATION OF HIGH-RESOLUTION CARDIAC HISTOANATOMICAL AND MRI IMAGES," Biomedical Imaging: From Nano to Macro, 2007. ISBI 2007. 4th IEEE International Symposium on , vol., no., pp.572-575, 12-15 April 2007 doi: 10.1109/ISBI.2007.356916 URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4193350&isnumber=4193197>
- [24] MEEI-LING JAN; KEH-SHIH CHUANG; GUO-WEI CHEN; YU-CHING NI; CHEN, S.; CHIH-HSIEN CHANG; JAY WU; TE-WEI LEE; YING-KAI FU; , "A three-dimensional registration method for automated fusion of micro PET-CT-SPECT whole-body images," Medical Imaging, IEEE Transactions on , vol.24, no.7, pp.886-893, July 2005 doi: 10.1109/TMI.2005.848617 URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1461524&isnumber=31446>
- [25] MESKINE, F, EL MEZOUAR, M, CH, TALEB, N. A rigid registration based on the nonsubsampled contourlet transform and genetic algorithms. Sensors. Algeria. 2010. [cit. 2011-11-14]. Dostupný z WWW: [www.mdpi.com/1424-8220/10/9/8553/pdf](http://www.mdpi.com/1424-8220/10/9/8553/pdf)
- [26] MOLGA, Marcin, SMUTNICKY, Czeslaw. Test functions for optimization needs. 2005. ] [cit. 2011-12-12]. Dostupný z WWW: < [www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf](http://www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf) >
- [27] NEJAD, A.G.; AYATOLLAHI, A.; , "Genetic algorithm as the main optimizer for medical image registration," Biomedical Engineering (ICBME), 2010 17th Iranian Conference of , vol., no., pp.1-3,

- doi: 10.1109/ICBME.2010.5704966  
 URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5704966&isnumber=5704913>
- [28] NISSANOV, J.; EILBERT, J.; TRETIK, O.J.; GEFEN, S.; SCHREMMER, S.; GUSTAFSON, C.; BERTRAND, L.; , "3D atlases - bridges between neurogenomics and neuroanatomy," *Biomedical Imaging*, 2002. Proceedings. 2002 IEEE International Symposium on , vol., no., pp. 357- 360, 2002  
 doi: 10.1109/ISBI.2002.1029267  
 URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1029267&isnumber=22111>
- [29] POŠÍK, P. Paralelní genetické algoritmy. 2001. Diplomová práce. Dostupné z WWW: < [labe.felk.cvut.cz/~posik/dipl/Diplomka.html\\_Toc504815302](http://labe.felk.cvut.cz/~posik/dipl/Diplomka.html_Toc504815302)>
- [30] RONKOVÁ, P. Kriteriační funkce pro registraci obrazů sítnice. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 51 s., 2 příl.
- [31] ROUET, J-M, JACQ, J-J, ROUX, CH. Genetic algorithms for robust 3-D MR-CT registration. *IEEE Transaction On Information Technology In Biomedicine*, Vol. 4, No. 2, 2000. [cit. 2011-11-16]. Dostupný z WWW: < [ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=845205](http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=845205)>
- [32] SEIXAS, F. L., OCHI, L. S., CONCI, A., SAADE, D.M. Image registration using genetic algorithms, *Proceedings of the 10<sup>th</sup> annual conference on Genetic and evolutionary computation (GECCO '08)*, Maarten Keijzer (Ed.). ACM, New York, NY, USA, 1145-1146.
- [33] SILVA, L.; BELLON, O.R.P.; BOYER, K.L.; , "Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms," *Pattern Analysis and Machine Intelligence*, *IEEE Transactions on* , vol.27, no.5, pp.762-776, May 2005  
 doi: 10.1109/TPAMI.2005.108  
 URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1407879&isnumber=30528>
- [34] SUCHOMEL, O, HYHLÍK, T. Optimalizační metody v CFD – diferenciální evoluce. [cit. 2011-10-25]. Dostupný z WWW: <[www.suchomel.org/files/Optimalizace-v-CFD.pdf](http://www.suchomel.org/files/Optimalizace-v-CFD.pdf)>
- [35] ŠVENDA, P. Genetický algoritmus. Článek. [cit. 2011-11-14]. Dostupné z WWW: < [www.fi.muni.cz/~xsvenda/docs/GeneticAlgorithm2002.pdf](http://www.fi.muni.cz/~xsvenda/docs/GeneticAlgorithm2002.pdf) >
- [36] TVRDÍK, J. Evoluční algoritmy. Skripta. Ostrava. 2004. Dostupný na WWW: <[prf.osu.cz/doktorske\\_studium/dokumenty/Evolutionary\\_Algorithms.pdf](http://prf.osu.cz/doktorske_studium/dokumenty/Evolutionary_Algorithms.pdf)>
- [37] VALKOUN, P. Numerické metody nepodmíněné minimalizace. 2010. Diplomová práce. Dostupné z WWW: <[is.muni.cz/th/175087/prif\\_m/diplomova\\_prace.pdf](http://is.muni.cz/th/175087/prif_m/diplomova_prace.pdf) >
- [38] WEISE, T. *Global Optimization Algorithms – Theory and Application*, 2009 published by it-weise.de (self-published), Germany. 2009.
- [39] WINTER, S.; BRENDEL, B.; PECHLIVANIS, I.; SCHMIEDER, K.; IGEL, C.; , "Registration of CT and Intraoperative 3-D Ultrasound Images of the Spine Using Evolutionary and Gradient-Based Methods," *Evolutionary Computation*, *IEEE Transactions on* , vol.12, no.3, pp.284-296, June 2008  
 doi: 10.1109/TEVC.2007.907558  
 URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4387663&isnumber=4531586>
- [40] XIUQIN PAN; YONG LU; YUE ZHAO; XIAONA XU; YONGCUN CAO; , "Research on the Algorithm of Small Object Detection of Image Based on Genetic Algorithm," *Intelligent Information Technology Application Workshops*, 2008. IITAW '08. International Symposium on , vol., no., pp.420-423, 21-22 Dec. 2008  
 URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4731967&isnumber=4731858>
- [41] ZELINKA, Ivan, OPLATKOVÁ, Zuzana, ŠEDA, Miloš, OŠMERA, Pavel, VČELAŘ, František. *Evoluční výpočetní techniky principy a aplikace*. Praha : BEN-Technická literatura, 2009. ISBN 978-80-7300-218-3
- [42] ZITOVÁ, B, FLUSSER, J. Image registration methods: a survey. *Image Vision Computing*. 2003, vol. 21, s. 997-1000
- [43] ŽÁRA, Jiří, BENEŠ, Bedřich, SOCHOR, Jiří, FELKEL, Petr. *Moderní počítačová grafika: druhé, přepracované a rozšířené vydání kompletní průvodce metodami 2D a 3D grafiky*. Brno : Computer Press, a.s., 2004. s. 609. ISBN 80-251-0454-0
- [44] Genetické algoritmy. [cit. 2011-11-14]. Dostupné z WWW: < [www.sledujplanuj.cz/Clanek/10/Geneticke-algoritmy](http://www.sledujplanuj.cz/Clanek/10/Geneticke-algoritmy)>

## Slovník

Alela – forma genu, konkrétní symbol

Bolus - sousto

Fitness – ohodnocující funkce

Filopodia – výběžky buněk

Gen – pozice v chromosomu

Chromozom – řetězec genetické informace reprezentující znaky; u GA představuje daného jedince

Jedinec – nositel genetické informace

Mydriáza – rozšíření zornice

Populace – skupina všech jedinců

Potomek – jedinec vytvořený z rodičů

Rodič – jedinec vybraný selekcí, určený pro vznik jedinců

Simplex – geometrický útvar s počtem vrcholů o jedna větším než je dimenze jeho prostoru

Tetraedr – čtyřstěn, těleso tvořené čtyřmi rovnostrannými trojúhelníky

Videofluoroskopie – skiaskopické vyšetření polykacího aktu s vytvářením dynamického záznamu

## Seznam zkratk

CT – počítačová tomografie (computer tomography)

GA – genetický algoritmus

GUI – grafické uživatelské prostředí (grafical user interface)

HEK – buněčná linie lidských embryonálních buněk z ledvin; human embryonal kidney

HybGA – hybridní genetický algoritmus

MI – Mutua Information, kritériální funkce Vzájemná informace

NM – Nelder-Mead metoda

NMR – nukleární magnetická rezonance

NP – nastavení parametrů

PET – pozitronová emisní tomografie



PGA – paralelní genetické algoritmy

PMX – křížení s částečným přiřazením

PSO – optimalizační metoda rojení částic (partial swarm optimization)

SA, SIMSA – Metoda simulovaného žihání

SD – směrodatná odchylka

SSD – Sum of Squared Differences, kritériální funkce Suma rozdílů čtverců

## **Seznam příloh**

Příloha A – Obsah přiloženého CD

Příloha B – Uživatelské prostředí optimalizace dvou snímků

Příloha C – Grafy testovacích funkcí

Příloha D – Tabulky naměřených hodnot

## Příloha A - Obsah přiloženého CD

Název adresáře	Popis adresáře
Data ke grafům	Soubor s tabulkami
Obrázky a grafy	Soubory použitých a vytvořených obrázků a grafů
Text	Elektronická verze diplomové práce
Zdrojové soubory	Skripty a funkce vytvořené v prostředí Matlab (Version 7.5.0.342 (R2007b))

## Příloha B – Uživatelské prostředí optimalizace dvou snímků

**Maska**

Vyber obrázky

Vytvořit masku s názvem:  .mat  
(bez diakritiky a mezer)  
Před vytvořením zadej vzorkování!

Načíst masku

**Volba parametrů**

Velikost počáteční populace:

Počet iterací:

Meze posunu v ose x:

Meze posunu v ose y:

Meze úhlu:

Pravděpodobnost křížení:

Pravděpodobnost mutace:

**Další možnosti**

Vzorkování obrazů:  %

Rozostření



Žádné

Obou obrazů

Jednoho obrazu

Spust výpočet


**Načtená data**

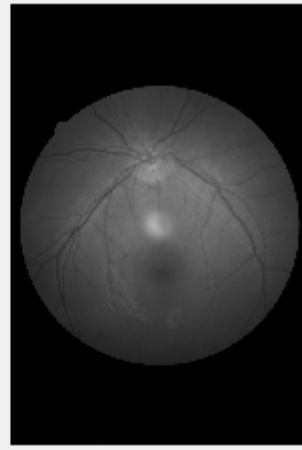
Velikost načtených dat: 2592 / 3888 [pixelů]

2592 / 3888

**Vymaskovaný referenční obraz**



**Slicované obrazy**



**Výsledky líčování**

Posun v ose x: 0

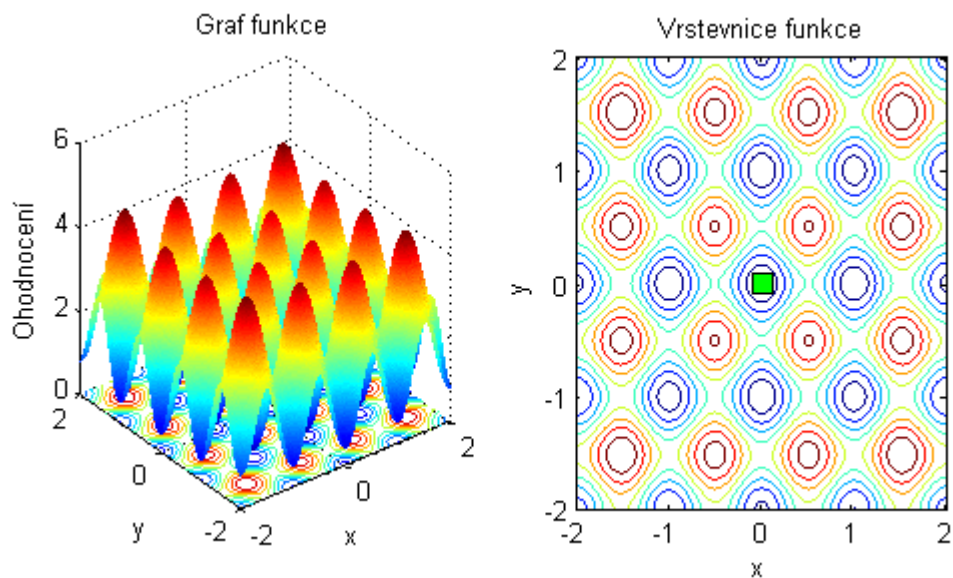
Posun v ose y: 0

Čas průběhu optimalizace: 282.948 sekund

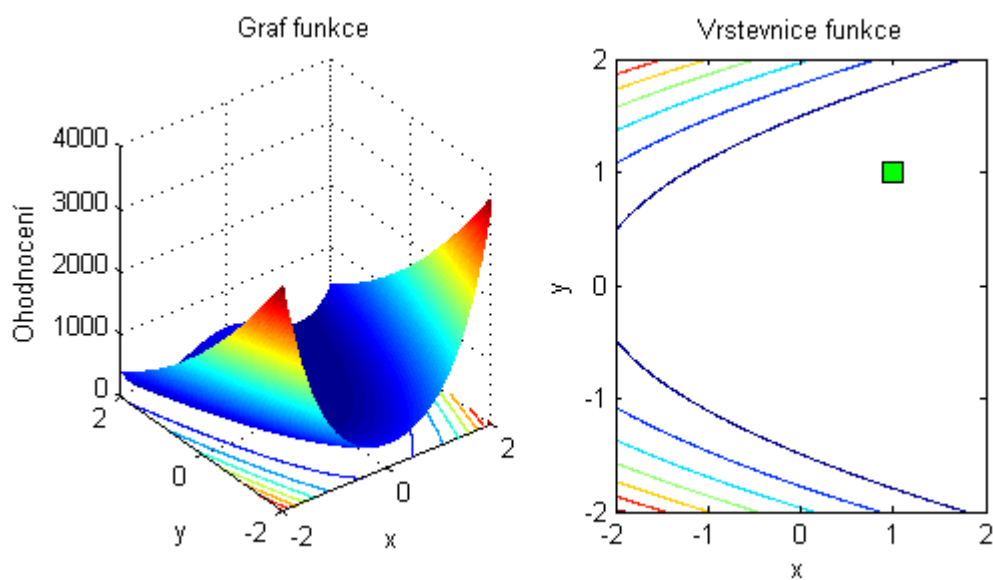
Změna úhlu: 0

## Příloha C – Grafy testovacích funkcí

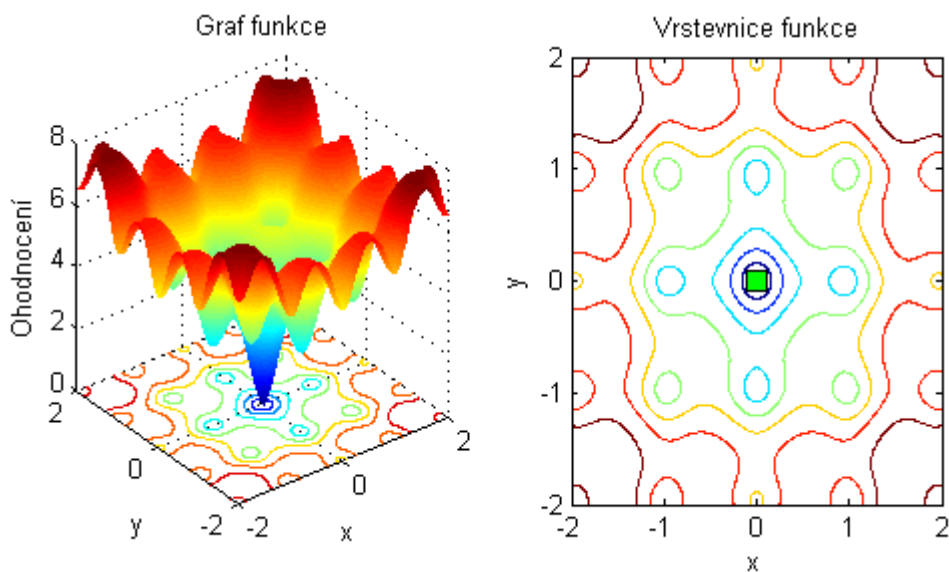
Graf Rastriginovy funkce v rozmezí  $x,y \in \langle -2, 2 \rangle$  vlevo; kontury funkce se znázorněným globálním minimem o hodnotě 0 na pozici  $x,y = \{0\}$  vpravo



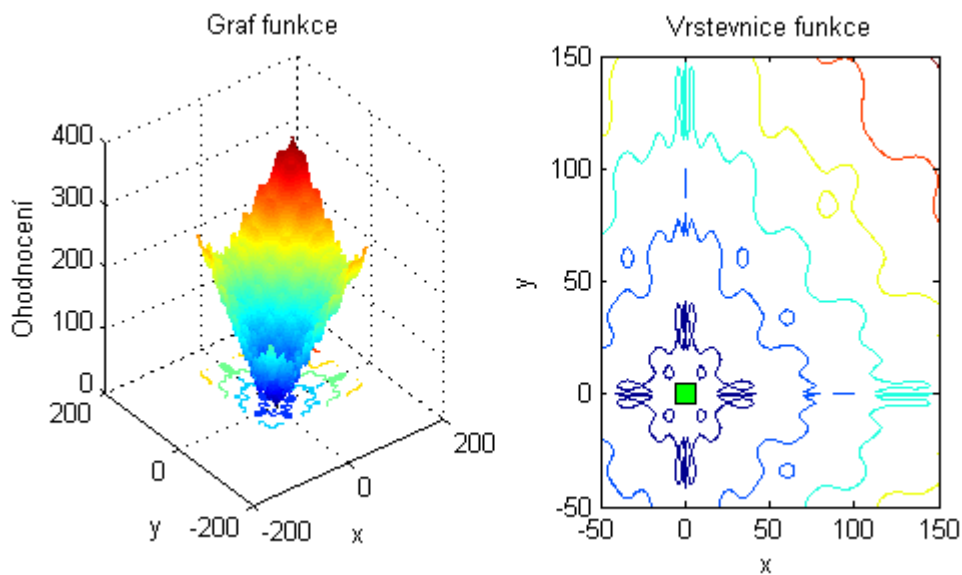
Graf Rosenbrackovy funkce v rozmezí  $x,y \in \langle -2, 2 \rangle$  vlevo; kontury se znázorněným globálním minimem hodnoty 0 na pozici  $x,y = \{1\}$  vpravo



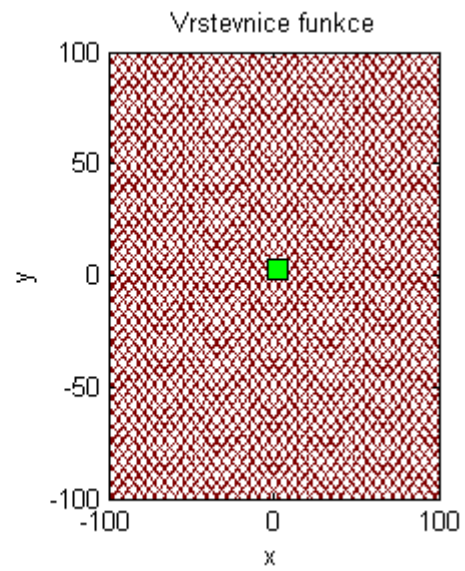
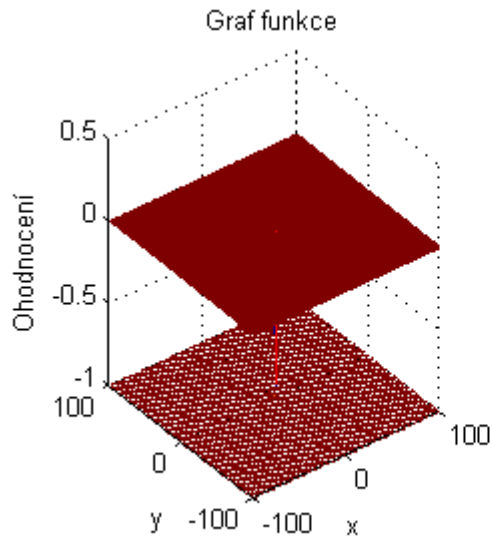
Graf Ackleyho funkce v rozmezí  $x,y \in [-2, 2]$  vlevo; kontury se znázorněným globálním minimem hodnoty 0 na pozici  $x,y = \{0\}$  vpravo



Graf podobný Ackleyho funkci s rozmezím  $x,y \in [-50, 150]$  vlevo; kontury se znázorněným globálním minimem s hodnotou 0 na pozici  $x,y = \{0\}$  vpravo



Vytvořený graf na bázi Ackleyho funkce s výrazným strmým hrotem a rozmezím  $x,y \in \langle -100, 100 \rangle$  vlevo; kontury funkce se znázorněným globálním minimem hodnoty 0 na pozici  $x,y = \{0\}$  vpravo



## Příloha D – Tabulky naměřených funkcí

Tabulka naměřených a vypočítaných hodnot při použití Rastriginovy funkce; prohledává se prostor  $x,y \in \langle -2, 2 \rangle$ , globální minimum se nachází v  $x,y = \{0\}$  a má hodnotu 0

	Časová náročnost	Dosažené ohodnocení	Odchylka ohodnocení od extrému	Výchylka v ose x	Výchylka v ose y	Vzdálenost od skutečného extrému
Fminsearch	0,199	0,099	0,099	0,000	0,995	0,995
	0,196	0,099	0,099	0,000	0,995	0,995
	0,212	0,199	0,199	0,995	-0,995	1,407
	0,199	0,099	0,099	0,000	-0,995	0,995
	0,201	0,099	0,099	0,000	-0,995	0,995
	0,196	0,099	0,099	0,000	0,995	0,995
	0,196	0,497	0,497	-0,995	1,990	2,225
	0,195	0,398	0,398	0,000	-1,990	1,990
	0,196	0,099	0,099	0,000	-0,995	0,995
	0,196	0,099	0,099	0,000	0,995	0,995
Ga	0,844	0,083	0,083	0,048	0,043	0,065
	0,846	0,255	0,255	1,028	0,953	1,401
	0,842	0,279	0,279	0,908	-0,042	0,909
	0,844	0,315	0,315	0,034	0,125	0,129
	0,840	0,136	0,136	0,999	0,043	1,000
	0,840	0,035	0,035	0,034	0,025	0,042
	0,842	0,224	0,224	0,076	0,972	0,975
	0,846	0,340	0,340	0,134	0,016	0,135
	0,841	0,056	0,056	0,030	0,044	0,053
	0,840	0,177	0,177	0,990	0,063	0,992
Simulannealbd	3,611	0,000	0,000	0,000	0,000	0,001
	3,335	0,100	0,100	0,995	-0,001	0,995
	5,515	0,099	0,099	0,995	0,000	0,995
	2,341	0,000	0,000	0,001	0,002	0,002
	4,588	0,000	0,000	0,000	0,000	0,001
	3,501	0,199	0,199	0,995	0,995	1,407
	3,685	0,000	0,000	0,000	0,000	0,000
	3,792	0,099	0,099	-0,995	0,000	0,995
	3,496	0,001	0,001	-0,006	0,000	0,007
	2,609	0,000	0,000	0,000	-0,003	0,003
GeneticAlgorithm	0,275	0,000	0,000	0,000	0,000	0,000
	0,278	0,100	0,100	0,990	0,000	0,990
	0,275	0,000	0,000	0,000	0,000	0,000
	0,254	0,000	0,000	0,000	0,000	0,000
	0,274	0,000	0,000	0,000	0,000	0,000
	0,274	0,000	0,000	0,000	0,000	0,000
	0,276	0,000	0,000	0,000	0,000	0,000
	0,274	0,000	0,000	0,000	0,000	0,000
0,263	0,000	0,000	0,000	0,000	0,000	

**Tabulka naměřených a vypočítaných hodnot při použití Rosenbrockovy funkce;  
prohledává se prostor  $x,y \in (-2, 2)$ , globální minimum se nachází v  $x,y = \{1\}$  a má hodnotu 0**

	Časová náročnost	Dosažené ohodnocení	Odchylna ohodnocení od extrému	Výchylka v ose x	Výchylka v ose y	Vzdálenost od skutečného extrému
Fminsearch	0,202	0,000	0,000	1,000	1,000	0,000
	0,211	0,000	0,000	1,000	1,000	0,000
	0,237	0,000	0,000	1,000	1,000	0,000
	0,202	0,000	0,000	1,000	1,000	0,000
	0,214	0,000	0,000	1,000	1,000	0,000
	0,206	0,000	0,000	1,000	1,000	0,000
	0,207	0,000	0,000	1,000	1,000	0,000
	0,204	0,000	0,000	1,000	1,000	0,000
	0,207	0,000	0,000	1,000	1,000	0,000
	0,205	0,000	0,000	1,000	1,000	0,000
Ga	0,879	0,010	0,010	0,965	0,923	0,085
	0,915	0,023	0,023	0,992	0,970	0,031
	0,872	0,066	0,066	0,960	0,947	0,066
	0,859	0,000	0,000	0,995	0,992	0,009
	0,890	0,037	0,037	0,880	0,759	0,270
	0,957	0,029	0,029	0,857	0,725	0,310
	0,886	0,104	0,104	0,943	0,921	0,097
	0,865	0,010	0,010	0,979	0,949	0,055
	0,861	0,084	0,084	0,735	0,529	0,540
	0,856	0,051	0,051	0,945	0,915	0,102
Simulannealbnd	2,265	0,001	0,001	1,034	1,069	0,077
	3,721	0,016	0,016	1,128	1,271	0,300
	3,957	0,000	0,000	0,990	0,979	0,023
	2,486	0,104	0,104	0,868	0,783	0,254
	3,306	0,004	0,004	0,934	0,872	0,144
	2,305	0,004	0,004	1,065	1,134	0,149
	3,334	0,003	0,003	1,051	1,104	0,116
	2,313	0,045	0,045	0,788	0,620	0,436
	3,292	0,000	0,000	1,013	1,025	0,028
	3,482	0,004	0,004	0,940	0,883	0,132
GeneticAlgorithm	0,273	0,023	0,023	0,850	0,720	0,318
	0,271	0,023	0,023	0,850	0,720	0,318
	0,269	0,023	0,023	0,850	0,720	0,318
	0,279	0,023	0,023	0,850	0,720	0,318
	0,275	0,023	0,023	0,850	0,720	0,318
	0,251	0,023	0,023	0,850	0,720	0,318
	0,270	0,023	0,023	0,850	0,720	0,318
	0,269	0,005	0,005	0,940	0,880	0,134
0,272	0,023	0,023	0,850	0,720	0,318	



**Tabulka naměřených a vypočítaných hodnot při použití Ackleyho funkce; prohledává se prostor  $x,y \in \langle -2, 2 \rangle$ , globální minimum se nachází v  $x,y = \{0\}$  a má hodnotu 0**

	Časová náročnost	Dosažené ohodnocení	Odchylka ohodnocení od extrému	Výchylka v ose x	Výchylka v ose y	Vzdálenost od skutečného extrému
Fminsearch	0,195	5,382	5,382	-0,982	1,965	2,197
	0,200	2,580	2,580	0,952	0,000	0,952
	0,195	2,580	2,580	-0,952	0,000	0,952
	0,193	6,560	6,560	1,975	1,975	2,792
	0,194	5,382	5,382	-0,982	1,965	2,197
	0,200	4,884	4,884	1,959	0,000	1,959
	0,196	3,574	3,574	0,969	-0,968	1,370
	0,195	4,884	4,884	0,000	1,959	1,959
	0,194	5,382	5,382	0,982	1,965	2,197
	0,195	5,382	5,382	0,982	-1,965	2,197
Ga	0,883	1,156	1,156	0,173	0,035	0,177
	0,860	1,828	1,828	0,132	0,211	0,249
	0,851	0,642	0,642	0,064	0,095	0,114
	0,855	1,846	1,846	0,102	0,232	0,254
	0,855	2,069	2,069	0,286	0,039	0,289
	0,857	0,707	0,707	0,059	0,107	0,122
	0,850	0,543	0,543	0,092	0,043	0,102
	0,855	0,727	0,727	0,118	0,041	0,125
	0,854	0,265	0,265	0,052	0,031	0,060
	0,858	0,624	0,624	0,075	0,083	0,112
Simulannealbnd	2,084	7,481	7,481	1,397	1,736	2,228
	2,351	3,876	3,876	1,055	0,235	1,081
	2,029	2,186	2,186	-0,303	-0,040	0,306
	2,035	3,221	3,221	-0,064	1,099	1,101
	2,042	2,042	2,042	-0,692	-0,756	1,025
	2,052	7,312	7,312	-1,768	-1,345	2,222
	2,015	5,766	5,766	-1,072	-1,328	1,706
	2,019	3,317	3,317	0,378	-0,265	0,462
	2,503	0,000	0,000	0,000	0,000	0,000
	2,028	4,587	4,587	-0,680	-0,737	1,003
GeneticAlgorithm	0,277	0,000	0,000	0,000	0,000	0,000
	0,286	0,000	0,000	0,000	0,000	0,000
	0,241	0,000	0,000	0,000	0,000	0,000
	0,264	0,000	0,000	0,000	0,000	0,000
	0,255	0,000	0,000	0,000	0,000	0,000
	0,253	0,000	0,000	0,000	0,000	0,000
	0,280	0,000	0,000	0,000	0,000	0,000
	0,254	0,000	0,000	0,000	0,000	0,000
0,253	0,000	0,000	0,000	0,000	0,000	

**Tabulka naměřených a vypočítaných hodnot funkce podobné Ackleyho funkci;  
prohledává se prostor  $x,y \in \langle -50, 150 \rangle$ , globální minimum se nachází v  $x,y = \{0\}$  a má hodnotu 0**

	Časová náročnost	Dosažené ohodnocení	Odchylna ohodnocení od extrému	Výchylka v ose x	Výchylka v ose y	Vzdálenost od skutečného extrému
Fminsearch	0,197	38,696	38,696	3,794	34,118	34,329
	0,203	3,870	3,870	-3,794	0,000	3,794
	0,196	65,771	65,771	-3,794	60,604	60,723
	0,199	96,726	96,726	60,604	34,118	69,548
	0,196	30,960	30,960	-15,172	-15,172	21,456
	0,197	143,055	143,055	135,939	-3,794	135,992
	0,204	65,771	65,771	60,604	-3,794	60,723
	0,216	38,696	38,696	34,118	3,794	34,329
	0,199	235,877	235,877	94,578	135,939	165,603
	0,197	131,519	131,519	-34,118	94,578	100,543
Ga	0,878	4,746	4,746	0,080	0,018	0,082
	0,851	7,319	7,319	0,092	0,062	0,110
	0,846	9,077	9,077	0,107	0,086	0,137
	0,850	4,479	4,479	0,082	0,011	0,083
	0,846	15,868	15,868	0,159	0,203	0,258
	0,844	10,911	10,911	0,046	0,201	0,206
	0,844	4,458	4,458	0,060	0,031	0,068
	0,876	10,056	10,056	0,170	0,052	0,178
	0,854	8,883	8,883	0,185	0,015	0,186
	0,850	10,507	10,507	0,019	0,226	0,227
Simulannealbd	3,442	4,634	4,634	0,000	-3,324	3,324
	2,562	3,872	3,872	0,000	3,773	3,773
	2,929	3,942	3,942	0,000	-3,942	3,942
	3,008	7,741	7,741	-3,793	3,793	5,365
	2,319	7,741	7,741	-3,794	3,795	5,366
	2,594	3,876	3,876	-3,752	0,000	3,752
	3,024	7,741	7,741	-3,795	-3,795	5,367
	4,435	3,876	3,876	0,000	3,752	3,752
	4,082	3,878	3,878	0,000	3,746	3,746
	2,771	6,859	6,859	-0,062	3,773	3,773
GeneticAlgorithm	0,294	10,084	10,084	-4,000	-3,000	5,000
	0,279	19,513	19,513	4,000	-15,000	15,524
	0,272	4,009	4,009	-4,000	0,000	4,000
	0,284	4,009	4,009	0,000	-4,000	4,000
	0,284	10,084	10,084	-4,000	-3,000	5,000
	0,283	10,084	10,084	3,000	-4,000	5,000
	0,283	10,084	10,084	3,000	-4,000	5,000
	0,280	4,009	4,009	0,000	-4,000	4,000
0,280	15,504	15,504	-15,000	0,000	15,000	

**Tabulka naměřených a vypočítaných hodnot funkce podobné medicínským snímkům;  
prohledává se prostor  $x,y \in \langle -100, 100 \rangle$ , globální minimum se nachází v  $x,y = \{3\}$  a má hodnotu  
-0,9416**

	Časová náročnost	Dosažené ohodnocení	Odchylka ohodnocení od extrému	Výchylka v ose x	Výchylka v ose y	Vzdálenost od skutečného extrému
Fminsearch	0,208	0,000	0,942	86,272	-54,540	101,218
	0,192	0,000	0,942	82,465	-45,100	92,889
	0,191	0,000	0,942	70,730	65,951	92,466
	0,192	0,000	0,942	89,412	-93,491	129,528
	0,191	0,000	0,942	11,781	-65,281	68,843
	0,200	0,000	0,942	-1,849	-76,259	79,408
	0,192	0,000	0,942	-80,795	-82,255	119,541
	0,196	0,000	0,942	-64,905	62,069	90,001
	0,194	0,000	0,942	36,304	-53,103	65,244
Ga	0,192	0,000	0,942	-37,333	-93,273	104,380
	1,395	0,000	0,942	1,108	1,355	2,507
	0,860	0,000	0,942	0,933	1,192	2,746
	0,849	0,000	0,942	0,889	0,934	2,953
	0,848	0,000	0,942	1,378	1,000	2,575
	0,846	0,000	0,942	0,831	0,839	3,062
	0,848	0,000	0,942	1,183	1,031	2,680
	0,846	-0,214	0,728	2,926	2,183	0,820
	0,849	0,000	0,942	1,344	1,506	2,231
Simulannealbd	0,846	0,000	0,942	0,861	0,993	2,933
	0,846	0,000	0,942	1,123	1,194	2,605
	2,613	0,000	0,942	17,858	-18,807	26,387
	1,983	0,000	0,942	-91,793	-48,634	107,943
	1,983	0,000	0,942	-4,862	11,354	11,471
	1,977	0,000	0,942	-4,118	-9,523	14,404
	1,989	0,000	0,942	-72,884	82,949	110,228
	1,981	0,000	0,942	-0,940	18,314	15,812
	1,983	0,000	0,942	25,741	-77,899	84,034
GeneticAlgorithm	1,996	0,000	0,942	-50,851	-52,911	77,627
	1,982	0,000	0,942	-67,475	37,893	78,640
	1,981	0,000	0,942	-0,472	-11,326	14,740
	0,285	-0,942	0,000	3,000	3,000	0,000
	0,282	0,000	0,942	11,000	-3,000	10,000
	0,283	-0,942	0,000	3,000	3,000	0,000
	0,279	0,000	0,942	30,000	-3,000	27,659
	0,280	0,000	0,942	11,000	-22,000	26,249
	0,266	-0,942	0,000	3,000	3,000	0,000
0,281	-0,942	0,000	3,000	3,000	0,000	
0,281	-0,942	0,000	3,000	3,000	0,000	
0,279	-0,942	0,000	3,000	3,000	0,000	