



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

OPTIMALIZAČNÍ METODA TRUST PRO REGISTRACI MEDICÍNSKÝCH OBRAZŮ

OPTIMIZATION METHOD BASED ON TRUST FOR IMAGE REGISTRATION

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

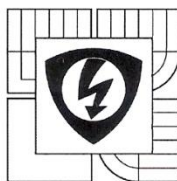
AUTOR PRÁCE
AUTHOR

Bc. LENKA PERNICOVÁ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. VRATISLAV HARABIŠ

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. Lenka Pernicová

Ročník: 2

ID: 106122

Akademický rok: 2011/12

NÁZEV TÉMATU:

Optimalizační metoda TRUST pro registraci medicínských obrazů

POKYNY PRO VYPRACOVÁNÍ:

1) Proveďte literární rešerši globálních optimalizačních metod, uvažujte jejich využití pro registraci medicínských obrazů. 2) Formulujte hlavní cíle práce směřující k návrhu a implementaci globální optimalizační metody založené na metodě TRUST pro registraci obrazu. 3) Navrhněte vhodnou optimalizační metodu založenou na metodě TRUST a rovněž navrhněte postup pro testování účinnosti a přesnosti optimalizační metody. 4) Optimalizační metodu implementujte v prostředí Matlab. 5) Navrženou metodu otestujte na dostatečném počtu dat, uvažujte různé extrémní případy, které mohou při registraci obrazů nastat. 6) Proveďte diskuzi navrženého řešení, diskutujte rovněž i výpočetní náročnost optimalizační metody v porovnání s dostupnými optimalizačními metodami.

DOPORUČENÁ LITERATURA:

- [1] CHEN, Y., BROOKS, R. R., IYENGAR, S. S., RAO, N. S. V., BARHEN, J.: Efficient global optimization for image registration, Knowledge and Data Engineering, IEEE Transactions on , vol.14, no.1, pp.79-92, 2002
[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

Tato diplomová práce se zabývá optimalizací pro registraci medicínských obrazů. Základem bylo seznámit se s registrací obrazů a důkladně prostudovat jednotlivé globální optimalizační metody, zvláště optimalizační metodu TRUST. Na základě teoretických poznatků bylo možné přejít k návrhu optimalizační metody založené na metodě TRUST pro registraci medicínských obrazů a k následné realizaci v programovacím prostředí Matlab. Vytvořený algoritmus je otestován na testovacích datech a srovnán s dalšími optimalizačními metodami jako je Simulované žíhání.

Klíčová slova

Registrace obrazu, optimalizace, globální minimum, metoda TRUST, tunelování, Matlab

Abstrakt

The aim of the thesis is optimization for a medical images registration. The basis is to acquaint with the images registration and to peruse component global optimization methods, especially an optimization method TRUST. After theoretic knowledge it is possible to proceed to a suggestion of an optimization method based on the TRUST method and to realize in the program setting MATLAB. Created algorithms has been tested on test data and compared with other optimization methods as Simulated annealing.

Keywords

Image registration, optimization, global minimum, method TRUST, tunneling, Matlab

PERNICOVÁ, L. *Optimalizační metoda TRUST pro registraci medicínských obrazů*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2012. 64 s. Vedoucí semestrální práce Ing. Vratislav Harabiš.

Prohlášení

Prohlašuji, že svou diplomovou práci na téma Optimalizační metoda TRUST pro registraci biomedicínckých obrazů 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 tohoto projektu 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 části druhé, hlavy VI. díl 4. Trestního zákoníku č.40/2009 Sb.

V Brně dne 18. 5. 2012

Podpis autora

Poděkování

Je mou milou povinností poděkovat panu Ing. Vratislavu Harabišovi, vedoucímu mé práce, za jeho trpělivost a návrhy vedoucí ke zlepšení kvality tohoto projektu.

V Brně dne 18. 5. 2012

Podpis autora

Obsah

Obsah	9
Seznam obrázků	11
Úvod	10
1 Registrace medicínských obrazů	11
1.1 Kriteriaální funkce	12
1.2 Geometrické transformace	13
1.3 Interpolace	13
2 Optimalizace	15
3 Globální optimalizační metody	17
3.1 Optimalizační metoda TRUST	17
3.1.1 Algoritmus TRUST	17
3.2 Další globální optimalizační metody	21
3.2.1 Genetické algoritmy (Genetic Algorithm)	21
3.2.2 Simulované žhání (Simulated Annealing)	22
3.2.3 Zakázané prohledávání (Tabu Search)	23
3.2.4 Metoda roje částic (Particle Swarm Optimization - PSO)	24
3.2.5 Metoda mravenčí kolonie (Ant Colony Optimization - ACO)	25
3.2.6 Memetické algoritmy (Memethic Algorithm)	26
4 Experimentální testování algoritmů	27
4.1 Testovací funkce	27
4.2 Testovací obrazce	34
4.3 Podmínky testování algoritmů	35
5 Hlavní cíle práce	37
6 Návrh a realizace algoritmu TRUST	38
6.1 Návrh TRUST algoritmu pro 1D	38
6.2 Realizace TRUST algoritmu pro 1D	39
6.2.1 Hlavní skript	39
6.2 Realizace TRUST algoritmu pro vícerozměrné případy	45
7 Testování algoritmu TRUST	47

7.1 Testování algoritmu na testovacích funkcích	47
7.2 Testování algoritmu na medicínských obrazech.....	52
7.2.1 Testování TRUST na obrazech z CT	52
7.2.1 Testování TRUST na obrazech z fundus kamery	55
Závěr	60
Použitá literatura.....	62

Seznam obrázků

Obr. 1 – Blokové schéma registrace obrazů [29]	12
Obr. 2 – Rozdělení algoritmů [24]	16
Obr. 3 – Metoda tunelování pro stanovení globálního minima [16]	18
Obr. 4 – Jeden iterační cyklus pro nalezení nového minima, kde černá křivka je původní $f(x)$ a červená křivka značí virtuální funkci $E(x, x^*)$	19
Obr. 5 – Genetický algoritmus [20]	22
Obr. 6 – Tabu search metoda [21]	24
Obr. 7 – První De Jongova funkce ($x=-2:0.1:2; x=y$)	28
Obr. 8 – První De Jongova funkce, vrstevnice s vyznačeným globálním minimem.....	28
Obr. 9 – Druhá De Jongova funkce ($x=-2:0.1:2; x=y$)	29
Obr. 10 – Druhá De Jongova funkce, vrstevnice s vyznačeným globálním minimem	30
Obr. 11 – Ackleyho funkce ($x=-2:0.1:2; x=y$)	31
Obr. 12 – Ackleyho funkce, vrstevnice s vyznačeným globálním minimem	31
Obr. 13 – Ackleyho funkce ($x=-100:5:100; x=y$)	32
Obr. 14 – Rastriginova funkce ($x=-2:0.1:2; x=y$)	33
Obr. 15 – Rastriginova funkce, vrstevnice s vyznačeným globálním minimem.....	33
Obr. 16 – Testovací obrazec 1, vhodný pro testování v 1D, např. ve směru x	34
Obr. 17 – Testovací obrazec 2.....	34
Obr. 18 – Vývojový diagram optimalizační metody TRUST pro 1D	38
Obr. 19 – Použitý testovací obrazec	40
Obr. 20 – Reflexe pro $d=2$	41
Obr. 21 – Redukce pro $d=2$	42
Obr. 22 – Virtuální funkce $E(x, x^*)$	43
Obr. 23 – Průběh kriteriální funkce nahoře a virtuální funkce dole pro 1. nalezené x^* označené v horním grafu černým křížkem	44
Obr. 24 – Transformační matice pro posuny v x a y, SSD	45
Obr. 25 – Srovnání výsledků TRUST a SA na První De Jongově funkci	48
Obr. 26 – Srovnání výsledků TRUST a SA na Druhé De Jongově funkci.....	49
Obr. 27 – Srovnání výsledků TRUST a SA na Ackleho funkci.....	50

Obr. 28 – Srovnání výsledků TRUST a SA na Rastriginově funkci	51
Obr. 29 – Obraz referenční	52
Obr. 30 – Obraz pohyblivý	52
Obr. 31 – Obraz zatížený šumem s Gaussovým rozložením, rozptyl 0.01	53
Obr. 32 – Závislost času potřebného pro nalezení extrému na velikosti šumu.	55
Obr. 33 – Obraz referenční	56
Obr. 34 – Obraz pohyblivý	56
Obr. 35 – Obraz znehodnocený šumem Pepř a sůl, hustota 0.01	57

Úvod

Registrace obrazů je důležitou součástí medicíny, jejíž výsledkem může být porovnání dvou a více obrazů, nalezení určitých struktur, sledování vývoje struktur (nádory) či kontrola léčby, popřípadě nalezení nových informací o pozorovaném objektu. Při zpracování medicínských obrazů se dbá na kvalitu. Ta nám určuje, jak dobře je možné lokalizovat patogenní oblasti, vykonat potřebné operace, aj. Důležitou cestou v medicíně je tedy neustálé zdokonalování algoritmů pro registraci obrazů.

Klíčovou roli v procesu registrace obrazů hraje optimalizace. Jejím úkolem je nalezení vhodné geometrické transformační funkce, při níž bude míra podobnosti obrazů maximální. Pro optimalizaci medicínských obrazů jsou hojně využívány metody jako Genetické algoritmy, Simulované žíhání či Zakázané prohledávání. Medicínská technika jde však stále dopředu, a proto jsou hledány stále nové a nové metody optimalizace.

Metodu TRUST původně představil Cetin a kolektiv [7] v roce 1993. Jedná se o metodu, která pro nalezení globálního extrému využívá virtuální funkce, která umožní tunelování do hodnot s nižší funkční hodnotou. Tato metoda bude řádně prostudována a na jejím základě bude navrhnout a vytvořen algoritmus v programovacím prostředí Matlab, který bude aplikován právě na medicínské obrazy.

1 Registrace medicínských obrazů

Pojem medicínské obrazy představuje obrazy (i vícerozměrné) pořízené z lékařských přístrojů, které pracují na základě různých fyzikálních zákonů a principů. Lze je rozdělit na dvě části zabývající se anatomií a funkcí. Mezi anatomické modalit lze zařadit CT (výpočetní tomografie), UZV (ultrazvuk), MRI (magnetická rezonance), aj. Do druhé části, která se zabývá funkcí, tedy ději v organismu ať patologickými nebo fyziologickými patří SPECT (jednofotonová emisní výpočtová tomografie), PET (pozitronová emisní tomografie) nebo také fMRI (funkční magnetické rezonance). Mimo tyto uvedené najdeme množství dalších modalit, jejichž výstupem jsou 2D či 3D obrazy, a proto klademe velký důraz jak na optimalizaci obrazů, tak na jejich celkovou registraci.

K základním problémům zpracování medicínských obrazů z různých lékařských přístrojů patří tedy registrace obrazu. Aby byla diagnostika obrazů na nejvyšší úrovni, je třeba realizovat co nejpřesnější registraci. Registrace obrazu je proces slícování dvou i více obrazů, které jsou vůči sobě posunuty. Posun může vzniknout při pohybu pacienta, který je snímán nebo také vlivem fyziologické či patologické deformace měkké tkáně ve snímaných obrazech.

Registrace má využití v mnoha oblastech, zde zmíním základní problémy, pro něž se registrace využívá [5]:

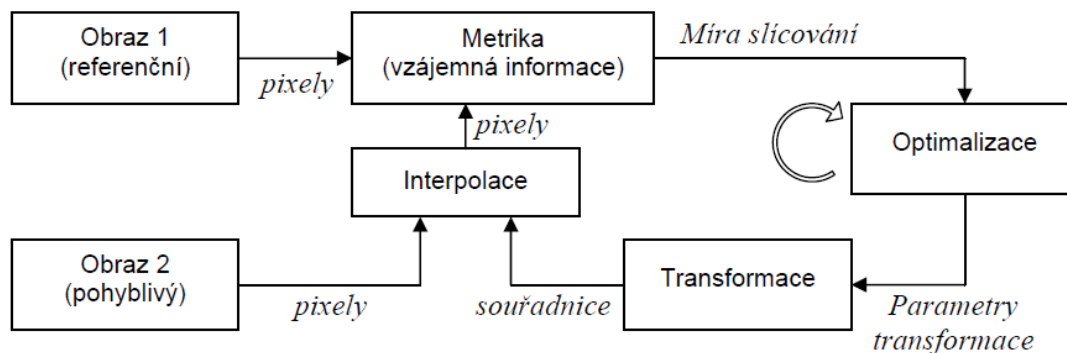
Multimodální registrace (Multimodal registration) – informace obrazů téže scény jsou snímány více senzory, získá detailnějších informací (obrazy různých modalit – např. PET snímá informaci o funkci a MRI obsahuje strukturní informace).

Hledání vzoru (Template registration) – v obraze hledáme předem známé struktury (např. komory mozku na MRI obraze), to je následně využito pro srovnání nalezené struktury s modelem.

Prostorová vizualizace (Viewpoint registration) – informace obrazů téže scény jsou snímány pod různými úhly (umožňuje stanovit třírozměrnou strukturu obrazu, snaha o panoramatický či prostorový snímek).

Sledování vývoje v čase (Temporal registration) – informace obrazů téže scény jsou snímány v různém čase a za různých podmínek, jedná se o snímání v určitých intervalech (využití při sledování nádorového onemocnění – růst nádoru).

Pro registraci je nutné nalezení vhodné geometrické transformace, která převádí oba obrazy (referenční a pohyblivý) do společného souřadnicového prostoru (obr. 1):



Obr. 1 – Blokové schéma registrace obrazů [29]

Na obr. 1 vidíme blokové schéma registrace obrazů. Nejprve zvolíme referenční a pohyblivý obraz a určíme jejich podobnost na základě zvolené kritériální funkce. Následně dochází k optimalizaci parametrů transformace a obraz je transformován a interpolován. Pomocí kritériální funkce se opět určí podobnost obrazů. V případě nedostačující podobnosti obrazů se celý proces registrace opakuje, jedná se tedy o iterativní proces, ve kterém se v každém kroku přibližujeme optimálnímu řešení.

1.1 Kritériální funkce

Míru podobnosti mezi referenčním a pohyblivým obrazem určíme pomocí kritériální funkce. Využíváme k tomu metody:

- Kritérium stochastické změny znaménka (SSC),
- Kritérium součtu absolutních hodnot rozdílů (SAVD),
- Suma rozdílů čtverců (SSD),
- Normovaný korelační koeficient (NCC),
- Vzájemná informace (IM), aj.

Suma rozdílů čtverců

Této metody bude následně využito ve vytvořeném algoritmu, a proto si ji podrobněji popíšeme.

Jedná se o jednoduchou metodu, která je založena na odečtení intenzit jednotlivých bodů obrazů. Předpokladem je, že k optimální registraci dojde, jestliže jsou obrazy identické.

Kritérium je definováno jako [3]:

$$SSD = \frac{1}{N} \sum_i^N |R(i) - L'(i)|^2 \quad (1.1)$$

kde N je počet shodných bodů, i je příslušný bod, R je vstupní obraz (referenční) a L' je obraz zpětně transformovaný (registrovaný).

Tato metoda se využívá u obrazů nebo oblastí, které jsou si velmi podobné. Jestliže je obraz zašuměn, dochází k selhání.

1.2 Geometrické transformace

Pod pojmem geometrická transformace při registraci medicínských obrazů rozumíme zpracování různých souřadných systémů do systému jednoho. Využívá se u toho geometrických transformací jako je translace, rotace, zkosení či změna měřítka. V rámci této práce se budeme zabývat zvláště translací a rotací (popř. změnou měřítka) [37].

Translace je dána vektorem posunutí $p = (x_t, y_t)$. Jedná se o posun se zachováním úhlů a vzdáleností bodů.

Rotace nám potočí referenční obraz vůči pohybovému o úhel α . Ve většině případů dochází k současné rotaci a translaci.

1.3 Interpolace

Interpolace je nedílnou součástí registrace obrazu, která se používá pro určení hodnot intenzity bodu ležícího mimo mřížku voxelů (zde intenzitu známe). Původní obraz je tedy v diskrétní podobě a jeho hodnoty jsou navzorkovány do uzlů mřížky. Po transformaci dojde ke

změně souřadnic a tím může dojít k tomu, že hodnoty se budou nacházet mimo tuto mřížku a tím může dojít ke ztrátě informací. Musíme proto využít technik interpolace pro výpočet souřadnic do uzlů mřížky. Při nevhodně provedené interpolaci nám může vzniknout ve výsledném obraze zkreslení či artefakty [26].

Nejjednodušší metoda interpolace je metoda nejbližšího souseda, kdy hledanou hodnotu určíme dle hodnoty nejbližšího známého bodu [11]. Mezi další interpolační metody patří aproximace jasové funkce, bilineární a bikubická transformace. Výběr metody je závislý na požadované rychlosti a přesnosti interpolace [4].

2 Optimalizace

Klíčovou roli v procesu registrace obrazů hraje optimalizace. Jejím úkolem je nalezení vhodné geometrické transformační funkce, při níž bude míra podobnosti obrazů maximální. Samotnou optimalizaci lze chápat jako hledání minima či maxima (míra vzájemné informace) kritériální funkce změnou hodnot stavových proměnných [31, 15]. Úspěšnou optimalizací docílíme úplného překrytí dvou obrazů (vstupního a referenčního).

Při řešení problému globální optimalizace tedy hledáme pro danou účelovou funkci

$$f: D \rightarrow R, D \subset R^d, \quad (2.1)$$

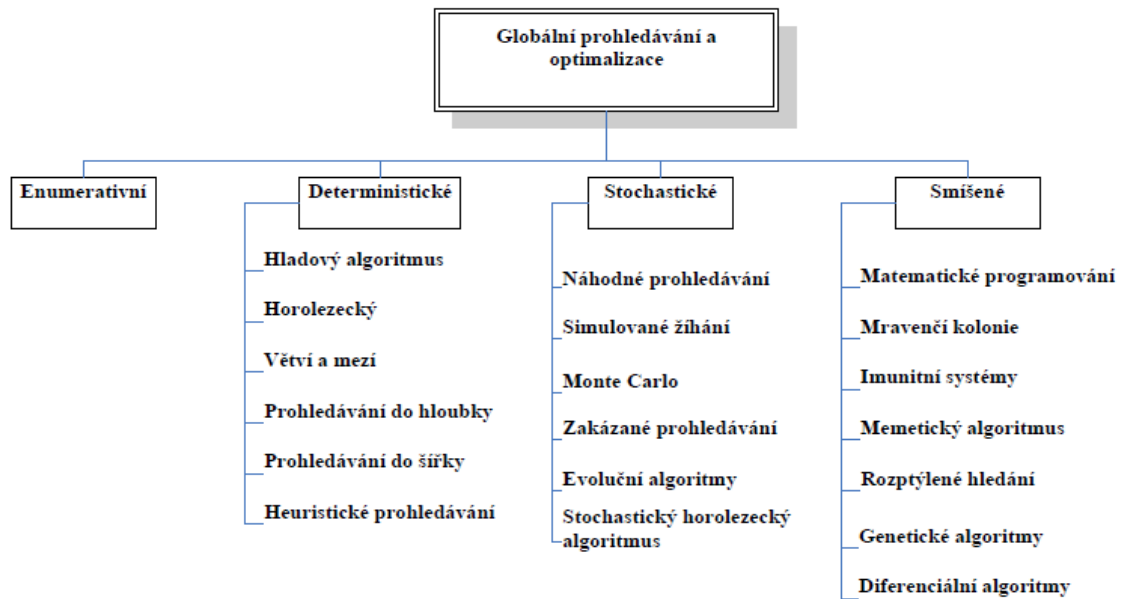
globální minimum x^* tak, aby

$$x^* = \operatorname{argmin}_{x \in D} f(x). \quad (2.2)$$

Pro jednoduchost uvažujeme D souvislou množinu ve tvaru

$$D = \prod_{i=1}^d \langle a_i, b_i \rangle, a_i < b_i, i = 1, \dots, d. \quad (2.3)$$

Při zpracování konkrétních obrazů je třeba aplikace vyhovující optimalizační funkce. Optimalizaci je možné rozdělit do čtyř základních skupin a to na metody enumerativní, stochastické, deterministické a smíšené (podrobněji níže, [24]). Rozdělení vidíme na *obr. 2*:



Obr. 2 – Rozdělení algoritmů [24]

Enumerativní metody vypočítávají všechny možné kombinace argumentů daného problému. Velmi časově náročné a nutný velmi malý počet hodnot, které mohou nabývat.

Deterministické metody se snaží nalézt minimum dané funkce pomocí prvních a druhých derivací. Použití je však pouze na hladké spojité funkce, což vede pouze k nalezení lokálních minim. Příkladem deterministické metody můžeme uvést Hill Climbing (Horolezecký algoritmus).

Stochastické metody zavádí do procesu optimalizace pravděpodobnost a náhodné jevy, což nám zajistí odstranění nedostatků deterministických metod (zpracování nespojitě funkce). Příklad stochastické metody je Simulované žihání.

Smíšené metody jsou kombinací algoritmů deterministických a stochastických (např. Genetické algoritmy). Dosahují velmi dobrých výsledků a jsou vhodné pro úlohy, kde je možno mnoha řešení.

3 Globální optimalizační metody

Pro správnou registraci medicínských obrazů je nutné využití globálních optimalizačních metod, které zabraňují uvíznutí algoritmu v lokálním minimu. Většinou se jedná se o metody s iterativním procesem, které ve většině případů nepotřebují znalost gradientu kriteriální funkce a velkou roli v nich hraje prvek náhody. To znamená, že se jedná o nedeterministické chování algoritmu, a tím může být ovlivněna kvalita řešení dané úlohy. Na druhou stranu výsledky v praxi ukazují efektivitu těchto algoritmů, a proto dochází ke stále větší popularitě těchto metod [24]. Dobrých výsledků dosahujeme pomocí aproximací, znalostí o problému, pomocí odhadů nebo zkoušením vhodných variant.

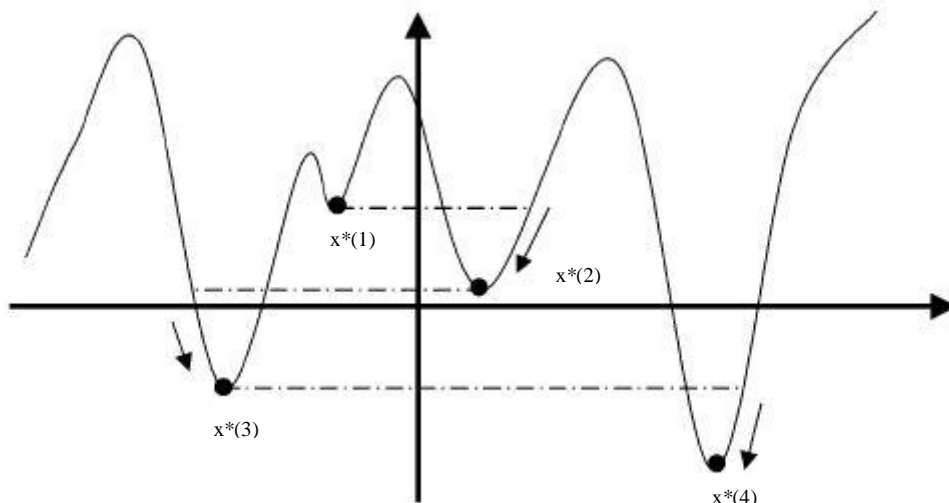
3.1 Optimalizační metoda TRUST

Výpočet registrace obrazu je velice náročný, a proto dochází k hledání nových optimalizačních globálních metod. Jako vzory pro registraci medicínských obrazů jsou hojně využívány metody Genetické algoritmy, Tabu Search neboli Zakázané prohledávání a Simulované žíhání. Stále však dochází k hledání nových a nových metod a technik.

Metodu TRUST původně představil Cetin a kolektiv [7]. TRUST metoda je založená na efektivním algoritmu globální optimalizace a je využívána ve vícerozměrných dimenzích, kdy je nalezeno lokální minimum a z tohoto minima se tuneluje do dalšího minima s nižší hodnotou (vyhneme se uvíznutí v lokálním minimu velmi rychle a účinně). TRUST ukazuje dobré výsledky konvergence ke globálnímu minimu a má dobře definované ukončovací kritérium [6].

3.1.1 Algoritmus TRUST

Algoritmus TRUST je sled cyklů, přičemž každý cyklus má dvě fáze – fáze nalezení lokálního minima a fáze tunelovací. Nacházíme se tedy ve výchozí pozici a v okolí hledáme první lokální minimum. Jakmile je nalezeno lokální minimum, předpokládáme, že toto minimum není globální, tudíž se snažíme tunelovat do nižších hodnot (dalšího minima s nižší hodnotou), (viz obr. 3).



Obr. 3 – Metoda tunelování pro stanovení globálního minima [16]

Popis algoritmu TRUST [10], [16], [23], [30] :

1. Celý algoritmus začíná v počátečním bodě $(X_{\min}, Y_{\min}, 0)$, tedy na hranici prohledávaného prostoru.
2. Pomocí vhodné metody nacházíme první lokální minimum, dále jen x^* .
3. Provedeme transformaci tak, že na původní funkci $f(x)$ se dostaneme do následujícího cíle pomocí virtuální funkce:

$$E(x, x^*) = E_{sub}(x, x^*) + E_{rep}(x, x^*), \quad (3.1)$$

kde

$$E_{sub}(x, x^*) = \log\left(\frac{1}{1+e^{-(f(x)-f(x^*))+a}}\right) \quad (3.2)$$

a

$$E_{rep}(x, x^*) = -\frac{3}{4}\rho(x-x^*)^{\frac{4}{3}}H(f(x)-f(x^*)). \quad (3.3)$$

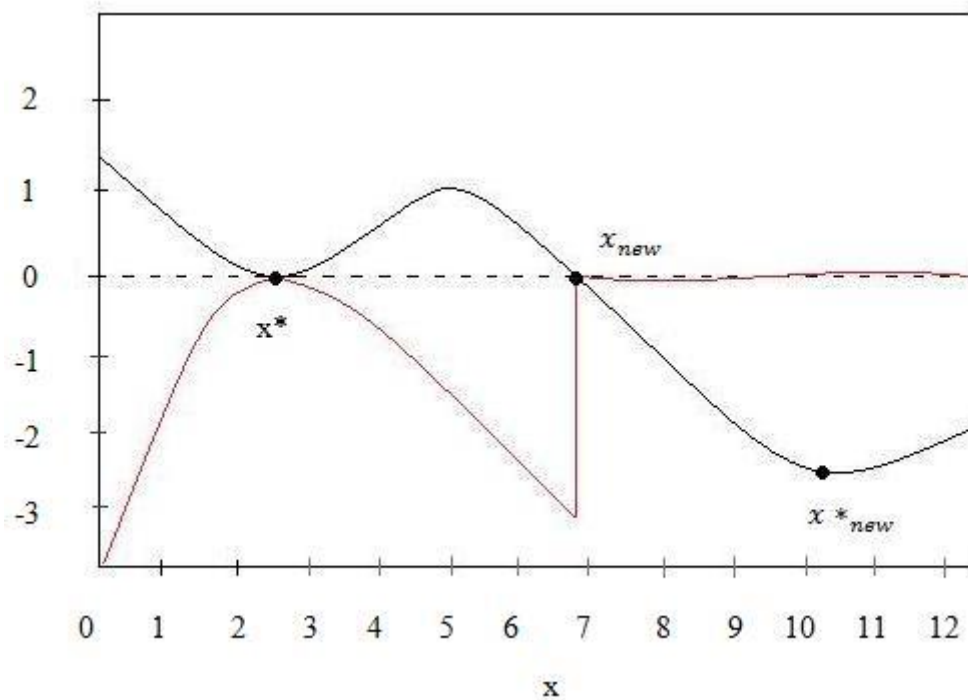
E_{sub} (subenergy tunneling function) se používá k omezení rozsahu funkce f , tzn., hodnot menších než funkční hodnota $f(x^*)$.

E_{rep} (repeller term) nám vyhledá další krok a zabrání uvíznutí v lokálním minimu.

ρ je pozitivní konstanta a H je Heavisideova funkce, což je nespojitá funkce, jejíž hodnota je nulová pro zápornou hodnotu argumentu a rovna jedné pro kladnou hodnotu argumentu.

Dohromady získáváme $E(x, x^*)$ (virtual objective function), což je dobře definovaná funkce, která nám nahradí původní funkci (rovn. 3.1).

4. Pomocí metody pro hledání lokálního minima na virtuální funkci $E(x, x^*)$ se dostáváme na další lepší pozici x_{new} , viz obr. 4.



Obr. 4 – Jeden iterační cyklus pro nalezení nového minima, kde černá křivka je původní $f(x)$ a červená křivka značí virtuální funkci $E(x, x^*)$

5. Vhodnou lokální optimalizační metodou nacházíme nové minimum x^*_{new} . Nové minimum x^*_{new} může být kromě lokálního minima původní funkce f také minimum globální.

Pokud nedojde ke splnění ukončovacího kritéria, vracíme se zpět na 3. krok, avšak nyní s novým vstupním lokálním minimem z předchozího kroku, tzn., nalezneme lokální minimum na původní funkci, vykreslíme si virtuální funkci, na té opět nalezneme lokální minimum, které nám dá bod v blízkosti lokálního minima na původní funkci a opět nalezneme lokální minimum na původní funkci a stále dokola, dokud není splněno ukončovací kritérium.

Ukončovací kritérium je dáno maximálním počtem iterací (tunelování) nebo je definováno jako překročení hranice prohledávaného prostoru, tzn., že v 1D prostoru vždy nalezneme globální minimum, protože dojde k důkladnému prohledání celého prostoru.

Algoritmus TRUST pro vícerozměrné případy funguje na stejném principu, jen je upraven vzorec pro virtuální funkci do tvaru:

$$E(\bar{x}, \bar{x}^*) = \log\left(\frac{1}{1+e^{-(f(\bar{x})-f(\bar{x}^*))+a}}\right) - \frac{3}{4}\rho \sum_{j=1}^n (x - x^*)^{\frac{4}{3}} H(f(\bar{x}) - f(\bar{x}^*)), \quad (3.4)$$

kde \bar{x} je vektor parametru 1 až n. Pro více parametrů nám ve vzorci přibyla suma pro 1 až n. Suma nám znázorňuje tedy sumu rozdílů jednotlivých parametrů.

V [16] byl TRUST zkoumán jak pro 1D případ, tak pro vícerozměrné případy. Bylo dokázáno, že pro 1D si algoritmus TRUST poradí výborně, to znamená, že prostor je vždy prohledán důkladně, takže globální extrém je vždy nalezen.

Pro vícerozměrné případy byl TRUST srovnán s algoritmem Elitní genetické algoritmy a Zakázané prohledávání. Konvergence ke globálnímu minimu není dokázána matematicky, avšak výsledky ukazují, že je srovnatelná s Elitními genetickými algoritmy. Výpočetní náročnost není u TRUST tak velká jak u Genetických algoritmů, navíc TRUST má velmi dobře stanovené ukončovací kritérium, jak již bylo zmíněno. Dále bylo testováno chování algoritmu v přítomnosti šumu, kde TRUST opět ukazuje skvělé výsledky.

Experimenty bylo dokázané, že TRUST metoda je výpočetně efektivnější než metoda Tabu Search nebo Genetické algoritmy. Např. Benchmarkovy testy zkoumají metody globální optimalizace a ukazují, že metoda TRUST je podstatně rychlejší. Tento algoritmus byl navržený pro paralelní analogové VLSI obvody pro podstatné vylepšení jejich rychlosti [16].

3.2 Další globální optimalizační metody

3.2.1 Genetické algoritmy (Genetic Algorithm)

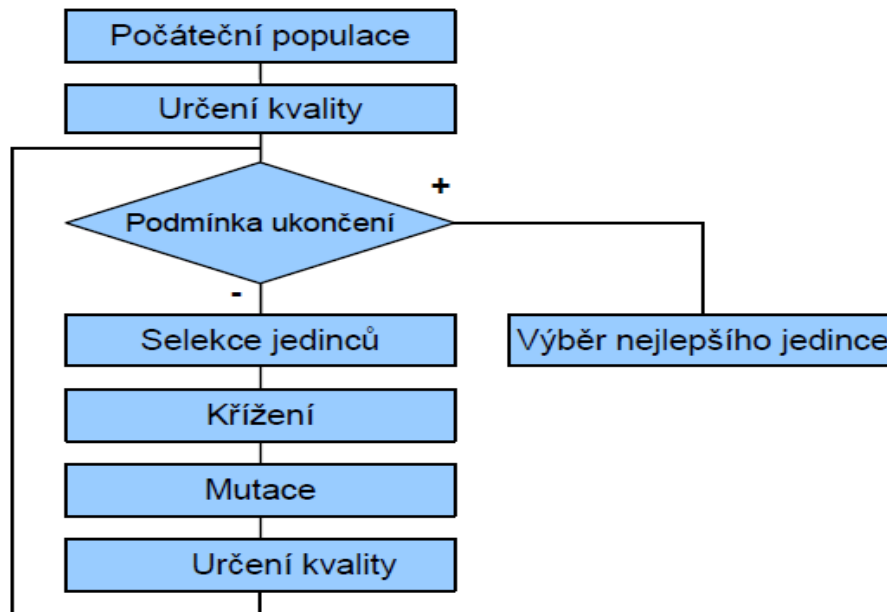
Vznik genetických algoritmů se datuje do 70. let dvacátého století. Základem této metody je Darwinova teorie o vývoji druhů. Ta říká, že děti dědí vlastnosti svých rodičů a silnější a přizpůsobivější jedinci lépe přežijí a plodí více potomků. Mírou přizpůsobení je tzv. „fitness“ funkce, tzn., že jedinci s větší fitness mají větší šanci přežít a rozmnožit se. V průběhu evoluce se dostáváme k vyšším fitness hodnotám [33].

Populace je množina různých řešení - jedinců, kde každý jedinec je reprezentován např. jako binární číslo. S takto zakódovanými jedinci můžeme provádět 3 operace [12], [18].

Selekce (Selection) – je operace, při které se každému jedinci přiřadí fitness funkce. Ta vyjadřuje míru zdatnosti, jak je jedinec schopný přežít a reprodukce. Aby se populace zlepšovala, je třeba provést selekci, to znamená, že vyřadíme jedince s malou fitness a tím zlepšíme zdatnost celé populace.

Křížení (Crossover) – je metoda, která navazuje na selekci a umožňuje jedincům si vyměnit část genetické informace. To umožňuje zvyšovat variabilitu řešení v populaci pomocí jejich částí [24]. Nejčastější je jednobodové, kdy se zvolí bod v chromozomu. Tento bod rozdělí chromozom na dvě části a ty se mezi potomky rozdělí. Křížením vzniknou dva noví jedinci. Musí se zvážít, zda se zanechají oba, nebo zda si náhodně vybereme jednoho z nich.

Mutace (Mutation) – jedná se o poslední operaci genetického algoritmu. Mutace slouží k udržení genetické variability a zabraňuje sklouznutí do lokálního optima [19]. Mutací může vzniknout informace, která v předchozí generaci nebyla či byla ztracena v průběhu předchozího vývoje [33]. V případě reprezentace binárním číslem se jedná o přehození 0 za 1 v náhodně vybraném chromozomu.



Obr. 5 – Genetický algoritmus [20]

Na obr. 5 vidíme výpočet genetického algoritmu. Nejdříve určíme počáteční populaci a následně cyklicky opakujeme selekci jedinců, křížení a mutaci. Proces opakujeme tak dlouho, dokud není splněna ukončovací podmínka, např. maximální počet generací, vypršení časového limitu pro nalezení optimálního řešení nebo dokud se zlepšuje kvalita jedinců v populaci.

Vylepšený genetický algoritmus je využit např. pro rigidní multimodální registraci 3D obrazů mozku [36].

3.2.2 Simulované žihání (Simulated Annealing)

Algoritmus simulované žihání je inspirován přírodními jevy a ději. Základem je chování termodynamických systémů, kdy po zahřátí tělesa docházím důsledkem řízeného ochlazování k formování krystalické mřížky v materiálu do stavu s co nejmenší energií [9, 34].

Tento algoritmus má iterační charakter. V každé iteraci vygeneruje náhodně bod a následně se náhodným mechanismem řízeným teplotou rozhodne, zda se posune do nově vygenerovaného bodu (dochází ke zlepšení stavu, lepší řešení) nebo zda zůstane do dalšího iteračního procesu v aktuálním místě.

Metoda Simulované žihání byla založena pro případy, kdy je hledán globální extrém v prostředí mnoha lokálních extrémů. Funguje na principu tzv. Boltzmannovy distribuce:

$$P(E) \sim e^{\frac{-E}{kT}}, \quad (3.5)$$

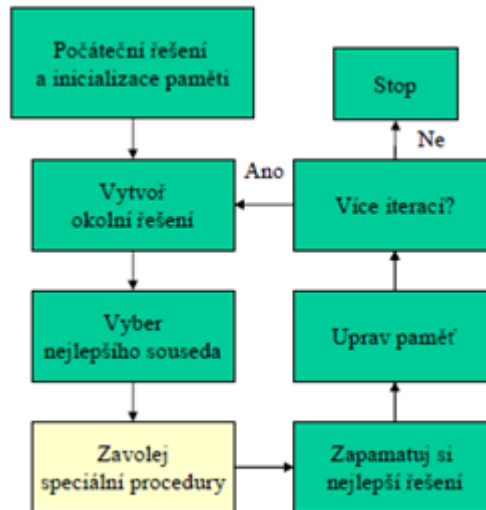
kde $P(E)$ udává pravděpodobnost, že systém přejde do energetického stavu E při teplotě T , k je Boltzmannova konstanta. V případě uvíznutí v lokálním extrému je přijato i aktuálně horší řešení, ale to jen s velmi malou pravděpodobností [1].

Simulované žihání je nejrozšířenější metoda pro registraci MRI obrazů [9].

3.2.3 Zakázané prohledávání (Tabu Search)

Další globální metodou je Tabu Search neboli Zakázané prohledávání. Jedná se o metodu, která je rozšířením metody Hill Climbing (Horolezecký algoritmus). Metoda Hill Climbing má tu nevýhodu, že vždy skončí pouze v lokálním minimu a globálního minima nikdy nedosáhne. Tabu Search rozšiřuje metodu Hill Climbing o paměť, která umožňuje vyhnout se uvíznutí v lokálním minimu. Do této paměti se ukládají stavy, které nemohou být znovu navštíveny po určité době (označeny jsou jako tabu). To znamená, že v dalším kroku jsou vždy vybírány ty neoptimálnější stavy, které ještě nebyly navštíveny (nebo byly navštíveny dávno). Tím je umožněn přeskok z lokálního do globálního optima [13].

Důležitým parametrem je velikost zakázaného (tabu) seznamu. Pokud se v tomto seznamu vyskytuje malý počet stavů, může dojít k zacyklení a nikdy nedosáhneme globálního minima. Pokud je naopak velikost seznamu příliš velká, může dojít k situaci, kdy dojde k “přeskoku” hlubokého údolí funkce. Tím přijdeme o lokální minimum, které se může ukázat také jako globální.



Obr. 6 – Tabu search metoda [21]

Na obr. 6 vidíme princip metody Tabu Search. Tato metoda byla pro využití při registraci medicínských obrazů kombinována s metodou genetických algoritmů. Adaptace algoritmu byla použita pro rigidní registraci obrazů z CT a laserového skenování. Optimalizovanou funkci představuje vzdálenost mezi detekovanými klíčovými body v obou obrazech [32].

3.2.4 Metoda roje částic (Particle Swarm Optimization - PSO)

Metoda roje částic je stochastická metoda založená na principu chování živočichů shlukujících se do roje či hejn (hejno ryb nebo ptáků, roj včel). Tato metoda byla vyvinuta v roce 1995 (J. Kennedy, R. C. Eberhart).

Princip si vysvětlíme na roji včel, které hledají na louce místo s největší hustotou rozkvetlých květin. Každá včela si pamatuje místo s největším počtem květin, které sama objevila a zároveň dostává informace od ostatních včel o jejich nejlepších polohách. Dále je včela informována o nejlepší poloze z celého roje. Včely stále přelétávají a hledají výhodnější pozice. V případě, že včela při přeletu z jednoho místa na jiné objeví výhodnější místo s větší hustotou květin, změní se její trajektorie podle nového místa. Na tomto principu je celé pole prozkoumáno rojem včel a skončí na místě s největší hustotou květin.

Při řešení technických problému touto metodou musíme určit [27]:

Prostor – výběr parametrů, které budeme optimalizovat a stanovíme rozsah řešení (maximální a minimální hodnoty)

Fitness funkce – je nutné ji určit nad celým prohledávaným prostorem. Pro kombinaci parametrů můžeme určit kvalitu řešení

Agent – určíme počet agnetů, které prostor prohledávají, jejich počáteční pozice a rychlost prohledávání. Pro každého agenta jsou určena stejná pravidla a pohybují se mezi osobním nalezeným maximem a maximem celého roje.

Osobní maximum – v každém bodě trajektorie agent srovnává aktuální hodnotu fitness s nejlepší hodnotou fitness osobně nalezenou. V případě větší hodnoty se osobní maximum přesouvá do aktuálního místa agenta.

Globální maximum – každý agent má informace o zbytku roje. Je určeno jedno globální maximum a agenti se pohybují mezi tímto globálním maximem a svým osobním maximem. V případě nalezení nového globálního maxima je vyslána informace, globální maximum je nahrazeno za nové a celý roj je přitahován k novému nalezenému místu.

Rychlost – rychlost změny výstupních parametrů v n-tém kroku ku n-1 kroku.

Algoritmus pracuje, dokud není nalezeno optimum nebo dokud neproběhne požadovaný počet iterací [27]. Využití metody roje částic najdeme hojně v robotice.

V roce 2002 bylo provedeno mnoho pokusů spojení Genetických algoritmů (GA) s Metodou roje částic (PSO). Bylo zjištěno, že nejlepších výsledků je dosaženo, pokud jako počáteční generaci GA zvolíme populaci optimalizovanou pomocí PSO [28].

3.2.5 Metoda mravenčí kolonie (Ant Colony Optimization - ACO)

Metoda mravenčí kolonie je jednou z moderních metod, která je úspěšná při řešení složitějších optimalizačních úloh. Vznik se datuje do 90. let 20. století.

Tento algoritmus je inspirován přírodou a to přesně mravenčí kolonií. Mravenci jsou drobní živočichové, skoro slepí a žijí ve velkých velmi dobře organizovaných koloniích. Jejich cílem je nalezení optimální cesty k potravě. Pohybují se po stopách feromonů, které po sobě mravenci zanechávají. Při rozcestí se rozhodují dle intenzity feromonů. Kde je intenzita větší, tudíž prošel mravenec vícekrát a tudíž je cesta kratší.

Pro účely optimalizace je vytvořen umělý mravenec, který má pozměněné vlastnosti pro zlepšení výsledků algoritmů. Dle skutečných mravenců je využito života v koloniích, feromonové stopy, naopak zavedenou vlastností je to, že množství feromonu je funkcí kvality nalezeného řešení.

Feromon je tedy vahou, která nám určuje zkušenosti ostatních mravenců při hledání slibného řešení. V algoritmu je zásadní i fakt, že pokud cesty nejsou používány dále, je feromon odpařován, což vede ke snížení vah u jednotlivých spojů, čímž se zvyšuje robustnost nalezení globálního extrému.

Algoritmus tedy na počátku každému mravenci přiřadí náhodné řešení, které je vylepšeno lokálním prohledáváním. Inicializuje se feromonová dráha a každý mravenec provede určitý počet kroků. Algoritmus se dostane buď do fáze zesilování (nejlepší řešení) nebo se naopak zesilování vypne. Pokud se jedná o nejlepší řešení, tak se v dalším kroku zvyšuje hladina feromonu. Proběhne několik iterací, pokud nedojde ke zlepšení řešení, tak se proces opakuje do nastolení některé z ukončovacích podmínek.

Optimalizace dle mravenčích kolonií je využita jako řídicí logika robotů schopných kooperace, další využití je například při barvení grafů [35].

3.2.6 Memetické algoritmy (Memethic Algorithm)

Memetické algoritmy jsou někdy označovány jako hybridní nebo paralelní genetické algoritmy. Tyto algoritmy mají více obecný koncept než genetické algoritmy a to z důvodu, že jsou kombinací genetických algoritmů a lokálním prohledáváním.

Princip algoritmu [24]:

Na začátku je vygenerována množina náhodných počátečních řešení. Poté je aplikováno na každé toto řešení lokální prohledávání, to nám řešení vylepší.

Poté jsou z množiny vybráni a rekombinováni dva rodiče, na výsledné chromozomy je opět aplikováno lokální prohledávání. Tento proces se opakuje do vybrání požadovaného počtu rekombinací.

Poté je vybráno nejlepší řešení a horší se odstraní.

Prohledávání je ukončeno při splnění některé z ukončovacích podmínek.

Tento algoritmus je využíván pro predikci proteinových struktur.

4 Experimentální testování algoritmů

Po realizaci algoritmu v Matlabu je nutné otestovat jeho účinnost a přesnost (postupně budeme sledovat algoritmus, jak správně pracuje při posunu v jednotlivých osách.). Dále budeme sledovat rychlost této metody, např. srovnáním s metodou Simulované žilání, které bude v rámci diplomové práce k dispozici. Na závěr můžeme také sledovat, jak se algoritmus TRUST chová ve více dimenzích (jak roste nebo klesá účinnost metody při přidávání dalšího rozměru, jako je např. změna měřítka).

4.1 Testovací funkce

Přesnost otestujeme pomocí obrazů nebo známých testovacích funkcí, u kterých známe správné řešení, tj. globální minimum. Jedná se o funkce, které mohou mít velké množství lokálních minim a jedno minimum globální, dále mohou mít pouze jeden extrém různé konvexnosti, atd. Mezi tyto funkce patří např. První De Jongova funkce, Druhá De Jongova funkce, Ackleyho funkce, Griewangkova funkce, Rastriginova funkce, Schwefelova funkce, Ranova funkce, Patologická funkce, aj. [14]. Nyní si popíšeme alespoň pět zmíněných testovacích funkcí a vykreslíme si je a jejich vrstevnice (globální minimum je označeno křížkem) v programovacím prostředí Matlab tak, aby byly dobře vidět jejich vlastnosti:

První De Jongova funkce

je jednoduchý rotační paraboloid, dle [33,22] definovaný jako

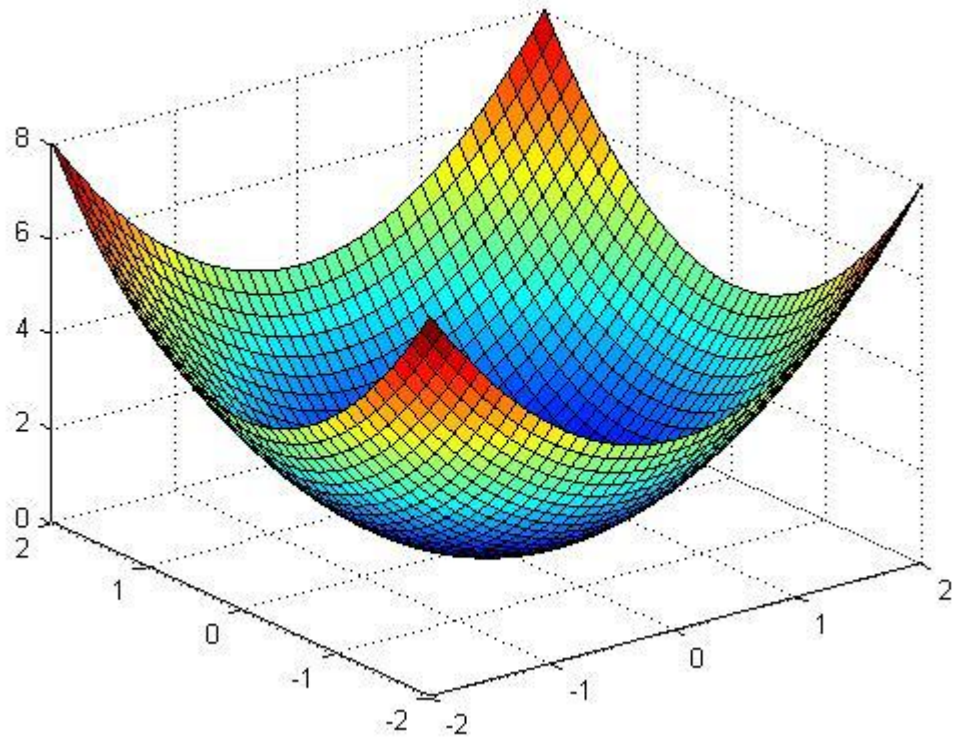
$$f(x) = \sum_{i=1}^d x_i^2. \quad (4.1)$$

Jedná se jednoduše o funkci, která má globální minimum v počátku soustavy souřadnic:

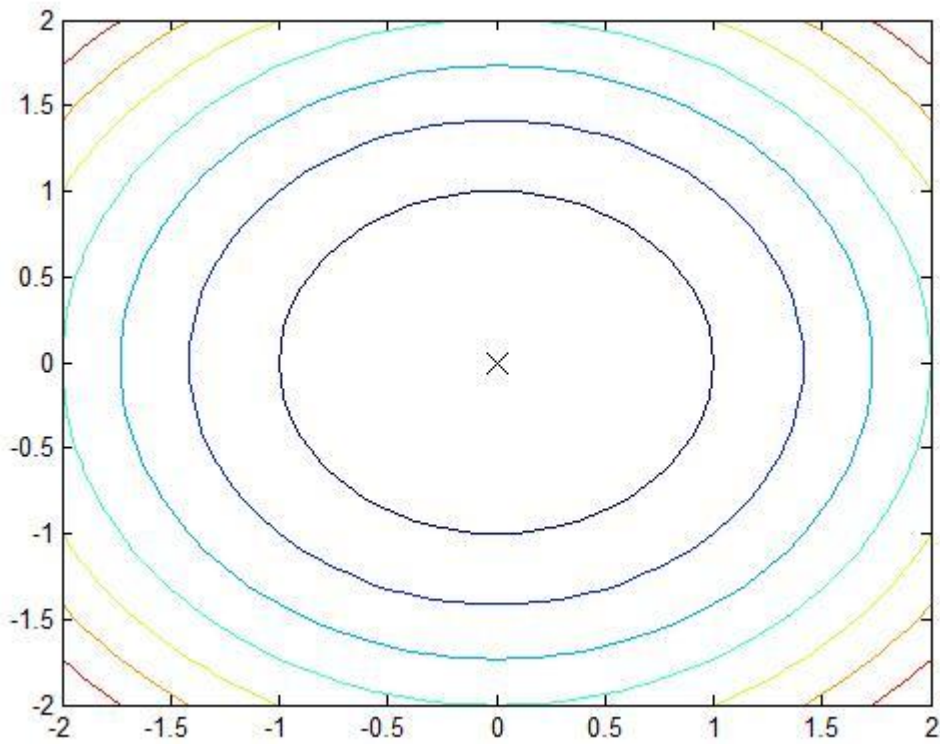
pro $d = 1$ je $x^* = (0)$,

pro $d = 2$ je $x^* = (0, 0)$,

pro $d = 3$ je $x^* = (0, 0, 0)$, atd., když $f(x^*) = 0$.



Obr. 7 – První De Jongova funkce ($x=-2:0.1:2$; $x=y$)



Obr. 8 – První De Jongova funkce, vrstevnice s vyznačeným globálním minimem

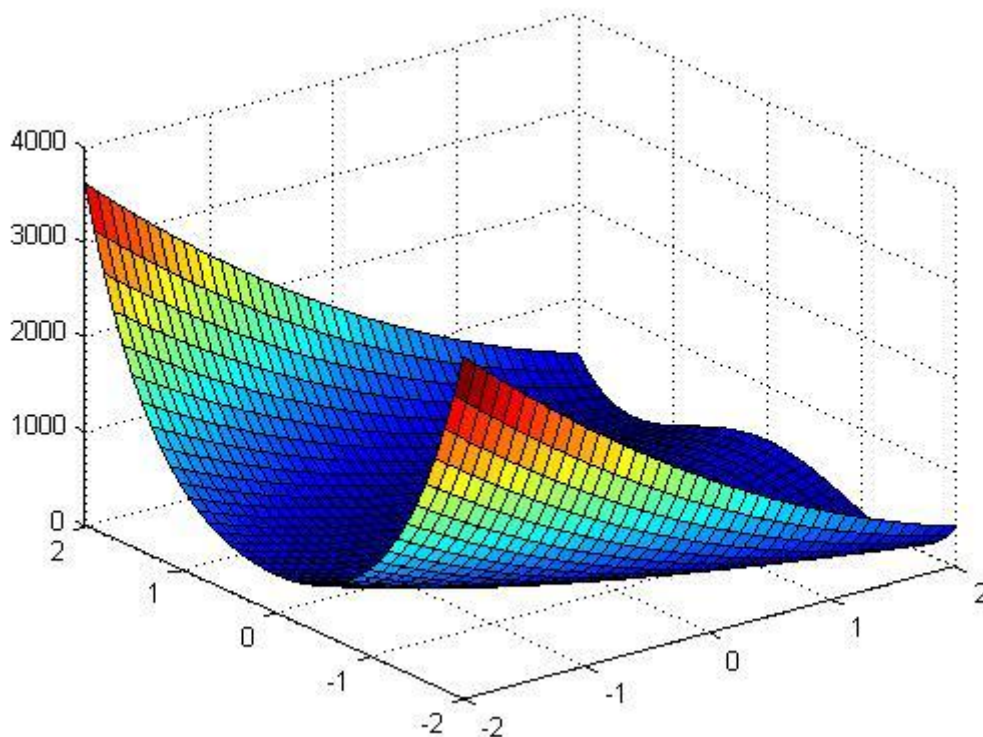
Pokud algoritmus selhává nebo pomalu konverguje, nezaslouží další pozornost. Funkce je vzhledem k jednomu minimu hojně využívána pro testování výkonnosti mnoha optimalizačních technik.

Druhá De Jongova funkce

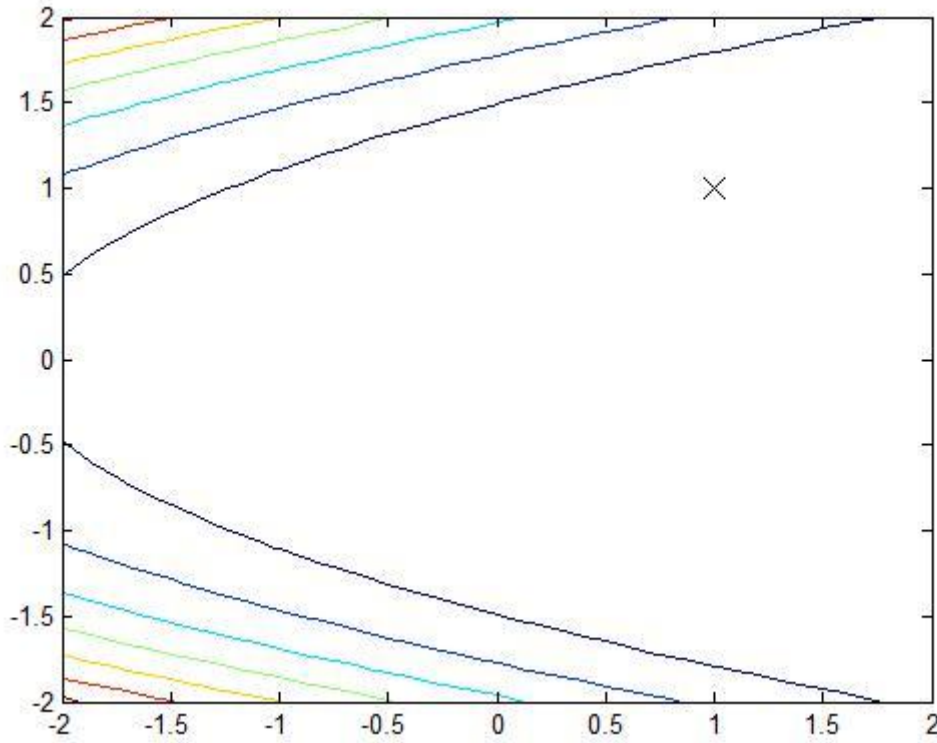
je známá také pod názvem Rosenbrockovo sedlo či Banánové údolí. Dle [33] je definována:

$$f(x) = \sum_{i=1}^{d-1} [100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2]. \quad (4.2)$$

Globální minimum leží v $x^* = (1, 1, \dots, 1)$, $f(x^*) = 0$. Potíže mohou nastat u iteračních algoritmů, které mohou své hledání ukončit předčasně. Důvodem je umístění minima v zahnutém údolí s malým spádem. Zajímavostí také je, že mimo globální extrém, obsahuje funkce také extrém velmi podobný hodnotou globálnímu extrému. Často využívaná testovací funkce [25], která je schopna úspěšně otestovat robustnost algoritmu.



Obr. 9 – Druhá De Jongova funkce ($x=-2:0.1:2$; $x=y$)

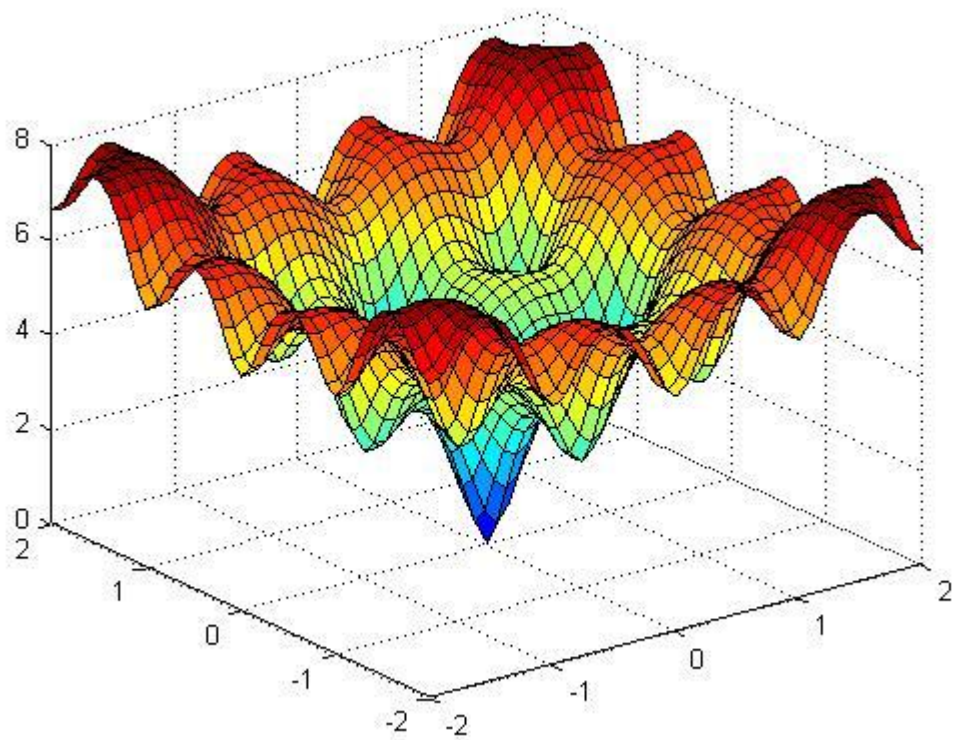


Obr. 10 – Druhá De Jongova funkce, vrstevnice s vyznačeným globálním minimem

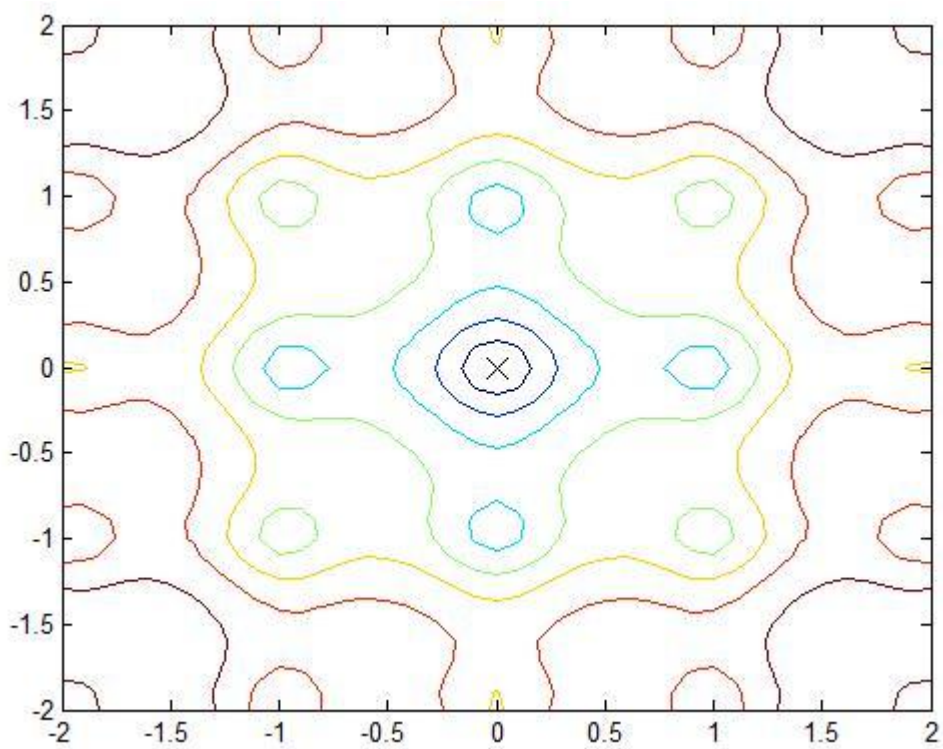
Ackleyho funkce

je často využívaná funkce, která obsahuje mnoho lokálních minim. Minimum najdeme v $x^* = (0,0, \dots, 0)$, $f(x^*) = 0$. Jedná se o středně obtížnou funkci, která je definována jako [33]:

$$f(x) = -20 \exp\left(-0.02 \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}\right) - \exp\left(\frac{1}{d} \sum_{i=1}^d \cos 2\pi x_i\right) + 20 + \exp(1). \quad (4.3)$$

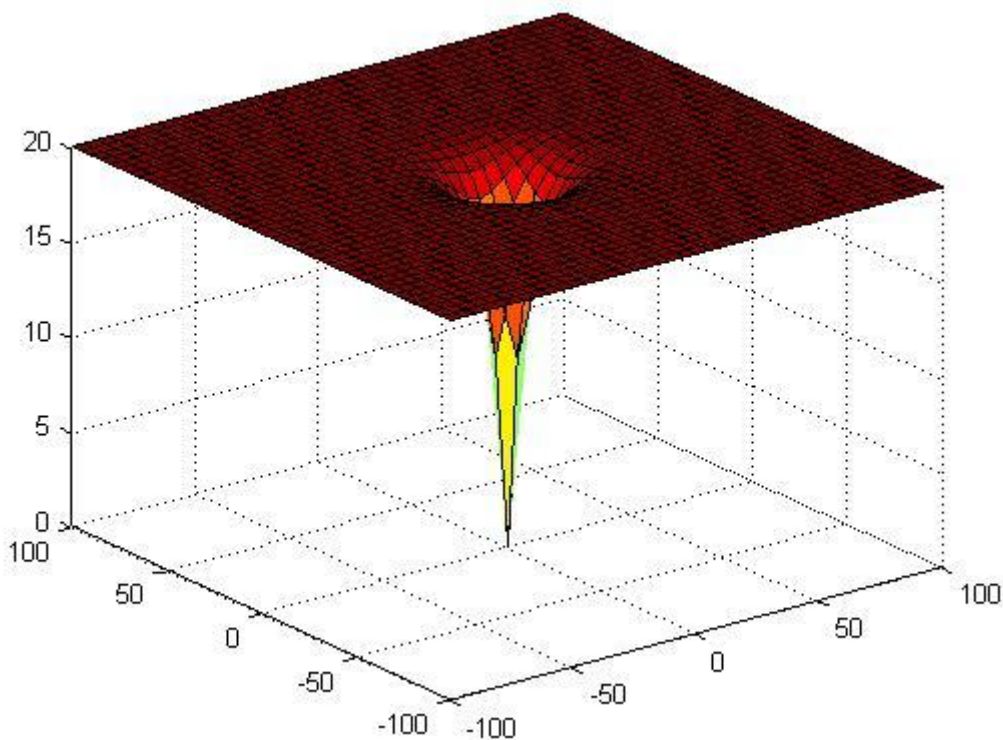


Obr. 11 – Ackleyho funkce ($x=-2:0.1:2$; $x=y$)



Obr. 12 – Ackleyho funkce, vrstevnice s vyznačeným globálním minimem

Při zvětšení rozsahu můžeme lépe pozorovat vlastnosti této funkce. Globální minimum vidíme ve středu a ve větší vzdálenosti od středu se nám funkce jeví jako rovina.



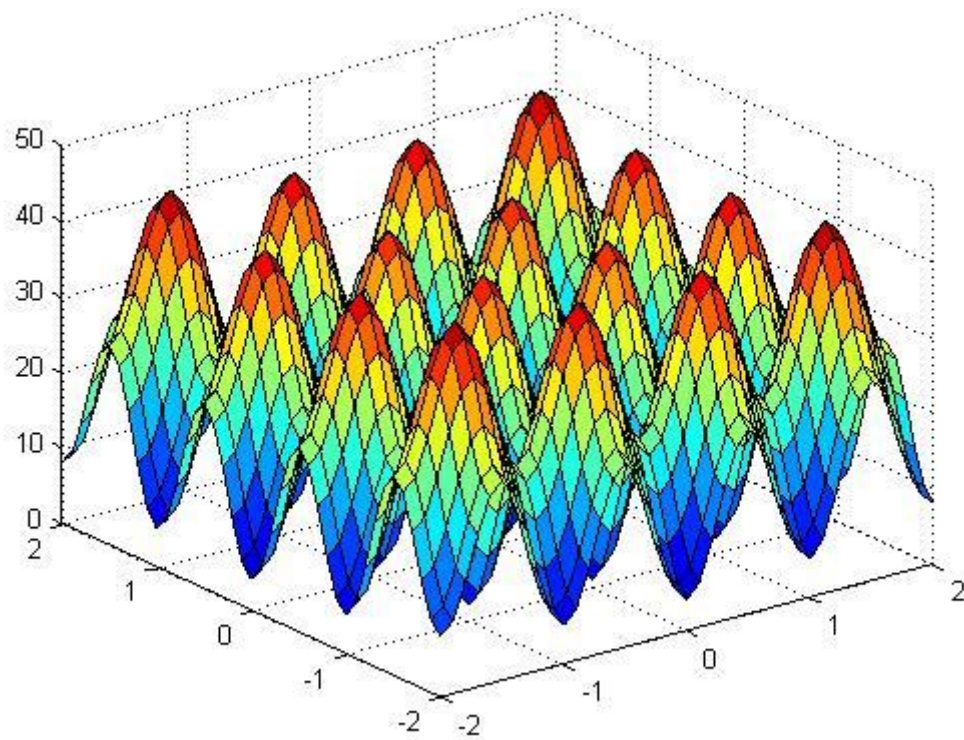
Obr. 13 – Ackleyho funkce ($x=-100:5:100$; $x=y$)

Rastriginova funkce

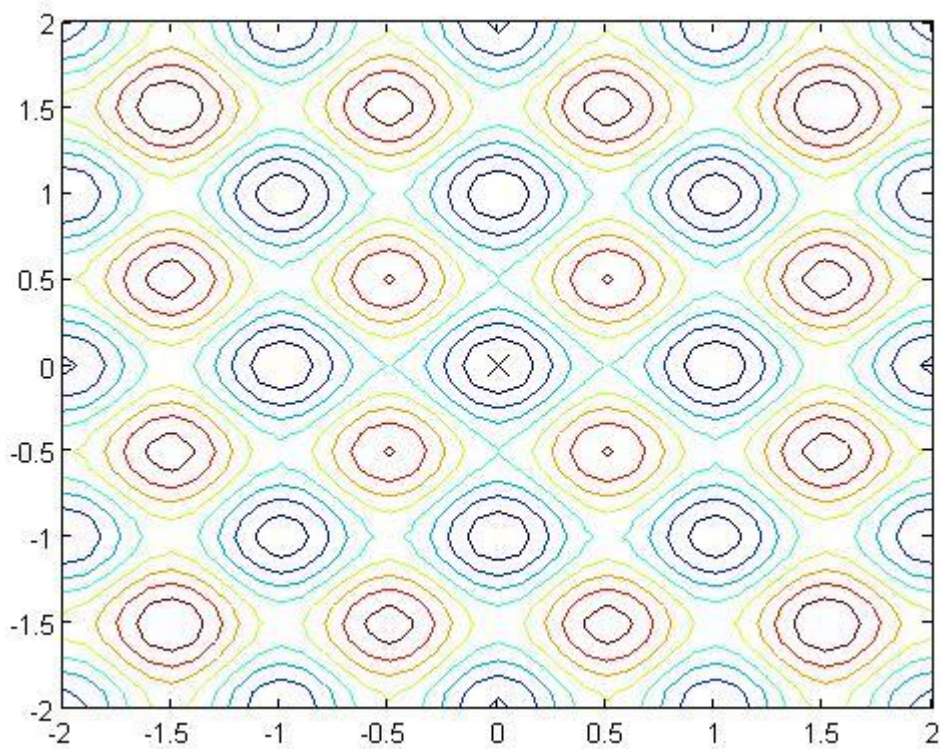
je často využívaná funkce obsahující spoustu lokálních minim a jedno minimum globální. Základem je První de Jongova funkce ke které je přidán kosinový průběh. Mnoho lokálních minim je však vidět pouze při malém rozsahu hodnot. Při velkém rozsahu hodnot se nám funkce jeví právě jako První de Jongova funkce.

Minimum najdeme stejně v $x^* = (0,0, \dots, 0)$, $f(x^*) = 0$. Nalezení minima je považováno za obtížnou úlohu globální optimalizace, která je definována jako [33]:

$$f(x) = 10d + \sum_{i=1}^d [x_i^2 - 10\cos(2\pi x_i)] \quad (4.4)$$



Obr. 14 – Rastriginova funkce ($x=-2:0.1:2$; $x=y$)

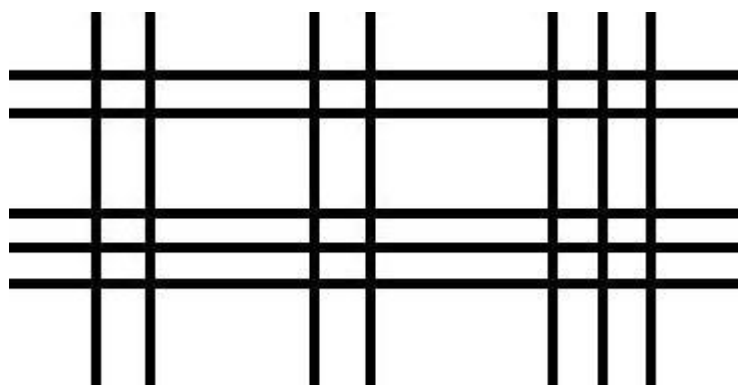


Obr. 15 – Rastriginova funkce, vrstevnice s vyznačeným globálním minimem

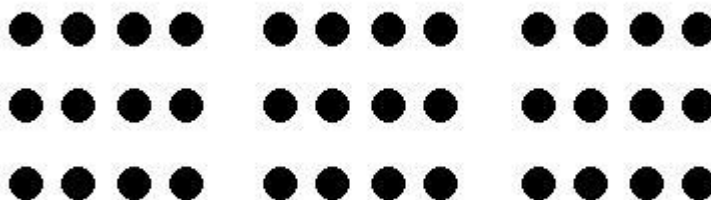
4.2 Testovací obrazce

Testovací obrazce jsou další možností pro testování algoritmu, tak jako testovací funkce. Testovací obrazec nám zobrazuje jak referenční obraz tak pohybový. Jedná se vlastně o dvě identické kopie, které vůči sobě posuneme. U testovacích obrazců využíváme existence bodu, kde hodnota slícování dvou obrazců je maximální, dojde tedy k dokonalému překryvu dvou obrazců.

Pro testování algoritmů jsou vhodné obrazce, které obsahují mnoho lokálních minim a jedno minimum globální. Nyní si některé vytvořené testovací obrazce zobrazíme:



Obr. 16 – Testovací obrazec 1, vhodný pro testování v 1D, např. ve směru x



Obr. 17 – Testovací obrazec 2

4.3 Podmínky testování algoritmů

Při testování dvou a více různých algoritmů je nutné dodržet stejné podmínky – vymezení prohledávaného prostoru, stejná podmínka ukončení prohledávání (např. stejný počet iterací) či nutnost testovat algoritmy na stejném počítači (různá rychlost procesorů). Sledujeme rychlost a spolehlivost nalezení globálního minima.

Podmínka ukončení algoritmu [17]:

$$f_{max} - f_{min} \leq \varepsilon,$$

f_{max} je největší funkční hodnota,

f_{min} je nejmenší funkční hodnota,

ε je práh ve většině případů volený jako 0,01.

Můžeme celkem dostat 4 možnosti ukončení algoritmu [17]:

Korektní ukončení

$$f_{max} - f_{min} \leq \varepsilon \text{ a } f_{min} \leq f_{near},$$

f_{near} hodnota, která určuje dostatečné přiblížení ke globálnímu minimu.

Jedná se o jediný typ úspěšného ukončení algoritmu. Ten se přiblížil ke globálnímu minimu před dosažením maximálního počtu iterací.

Pomalá konvergence

$$f_{min} \leq f_{near},$$

algoritmus konverguje, avšak je ukončen dosažením maximálního počtu iterací.

Předčasná konvergence

$$f_{max} - f_{min} \leq \varepsilon, \text{ ale } f_{min} > f_{near},$$

nebylo nalezeno globální minimum – uvíznutí v lokálním minimu.

Úplné selhání – dosažení maximálního počtu iterací, nepřiblížení se globálnímu minimu.

Spolehlivost nalezení globálního minima značíme R , což je poměr n_1 korektních ukončení v n nezávislých opakováních. Pro větší hodnoty n se jedná o normální rozložení R .

5 Hlavní cíle práce

Hlavní cíl – návrh vhodné globální optimalizační metody založené na metodě TRUST a její implementace do programovacího prostředí Matlab. Dále je nutné navrhnout postup pro testování účinnosti a přesnosti optimalizační metody.

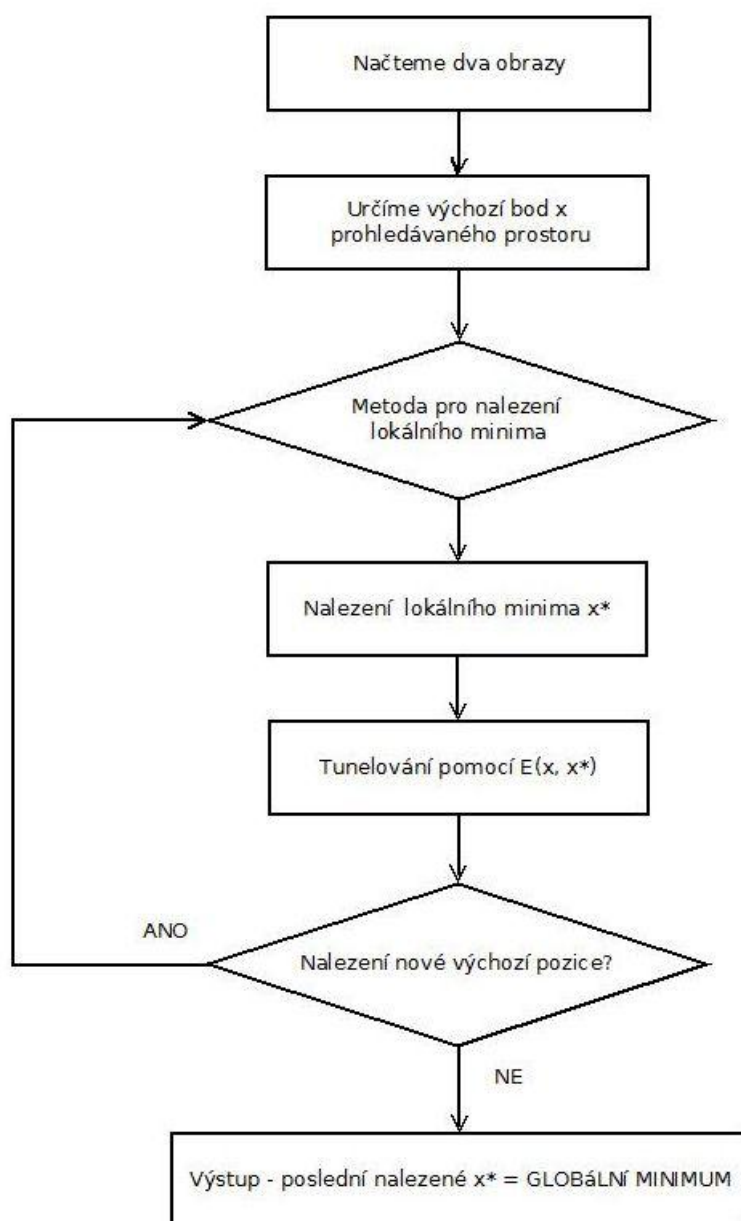
Pro splnění hlavního cíle, tedy návrhu optimalizační metody, je nutné vytvořit teoretická východiska. Nejprve je důležité důkladně prozkoumat jednotlivé globální optimalizační metody a zejména pochopit princip optimalizační metody TRUST pro registraci obrazu. Tento krok je celý důkladně probrán v první části této práce. Po té můžeme přejít k návrhu metody – vývojovému diagramu, který sestavíme na základě poznatků z teoretického základu.

V následující části práce implementujeme návrh 1D do programovacího prostředí Matlab, tuto metodu důkladně otestujeme na dostatečném počtu dat a případně provedeme potřebné úpravy. V případě, že algoritmus pracuje dobře, pak jej celý rozšíříme pro vícerozměrné případy. Pro registraci obrazů jej rozšíříme hlavně pro posuny v ose x a y . Algoritmus otestujeme na medicínských obrazech, popř. zvážíme situace či extrémy, které mohou při registraci obrazu vzniknout. Výslednou metodu srovnáme z hlediska výpočetní a časové náročnosti s dalšími metodami, které budou k dispozici.

6 Návrh a realizace algoritmu TRUST

6.1 Návrh TRUST algoritmu pro 1D

Návrh optimalizační metody TRUST zobrazíme pomocí vývojového diagramu. Do algoritmu vstupují obrazy načtené z 2 senzorů, obrazy snímané v určitém časovém intervalu po sobě či obrazy, které jsou sejmuté z různých úhlů.



Obr. 18 – Vývojový diagram optimalizační metody TRUST pro 1D

Výstup algoritmu nám udává hodnotu a pozici globálního minima. Dále můžeme sledovat počet cyklů, tedy počet nalezených lokálních minim neboli kolik tunelování do nalezení globálního minima proběhlo. Algoritmus je navržený na základě teoretického principu, který je uvedený v *kap. 3.1.1*.

6.2 Realizace TRUST algoritmu pro 1D

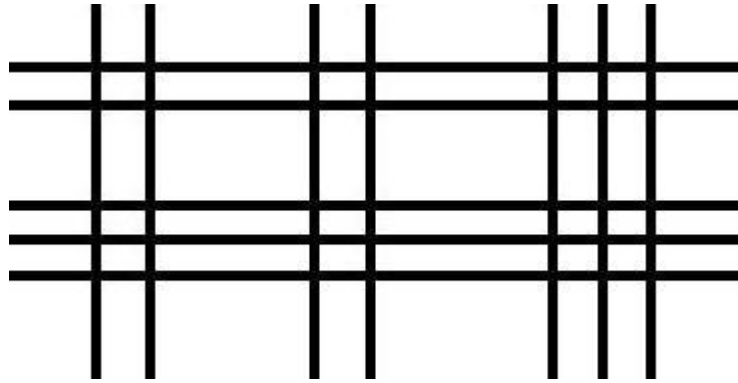
Návrh algoritmu TRUST pro 1D budeme nyní implementovat do programovacího prostředí Matlab. Celý program se skládá z hlavního skriptu a několika do něj vložených funkcí vytvořených v pomocných skriptech, pořípadě z funkcí, které Matlab Toolbox nabízí již vypracované. Nyní k jednotlivým částem hlavního skriptu.

6.2.1 Hlavní skript

Na začátku celého skriptu použijeme výrazů `clear all`, `clc`, `close all`. Pomocí těchto příkazů uzavřeme všechny předchozí funkce, zabráníme načtení proměnných ve Workspace, tím, že je vymažeme a otevřeme prázdnou obrazovku v Command Windows. Tímto by mělo být zabráněno jakémukoli střetu funkcí, programů či proměnných a je možno přejít k vytvoření samotného skriptu.

Načteme dva obrazy

Tento blok je nezbytnou součástí celé optimalizace. Do programu načteme dva obrazy, které jsou snímány z více senzorů nebo v krátkém intervalu po sobě. Pro realizaci je vhodné načíst testovací obraz a jeho kopii, kterou vůči původnímu obrazu posuneme. Tím získáme dva obrazy, které jsou vůči sobě posunuty na ose x. Tento postup volíme proto, abychom po vytvoření algoritmu mohli snadno zkontrolovat, zda funguje jak má. Obrazy načítáme pomocí příkazu `imread`.



Obr. 19 – Použitý testovací obrazec

Dalším příkazem objevující se v části načtení obrazů je `rgb2gray`. Ten nám zajišťuje převod barevného obrazu na obraz šedotónový. Tímto příkazem můžeme zrychlit chod programu potlačením rozměru barvy. Hlavní využití příkazu bude až u realizace návrhu pro vícerozměrné případy, kdy dojde k aplikaci algoritmu na medicínské obrazy.

Určíme výchozí bod prohledávaného prostoru

Jedná se o velmi jednoduchou součást celého skriptu. Výchozí bod optimalizačního algoritmu TRUST se volí z pravidla $(X_{\min}, Y_{\min}, 0)$. V našem případě, tedy pro 1D případ, volíme výchozí pozici (0) .

Metoda pro nalezení lokálního minima x^*

Pro nalezení lokálního minima, což je první fáze algoritmu založeném na optimalizační metodě TRUST, použijeme Simplexovou metodu popsanou níže dle [17].

Základem Simplexové metody je vygenerování simplexu S , tedy $d+1$ bodů v souvislé prohledávané oblasti D (v 1D se tedy jedná o úsečku, ve 2D již o tři body tvořící trojúhelník).

$$S = \{x_1, x_2, \dots, x_{d+1}\} \quad (6.1)$$

V simplexu S nalezneme body s nejvyšší (H) a nejnižší (L) funkční hodnotou, tedy:

$$x_H = \arg \max_{x \in S} f(x), \quad (6.2)$$

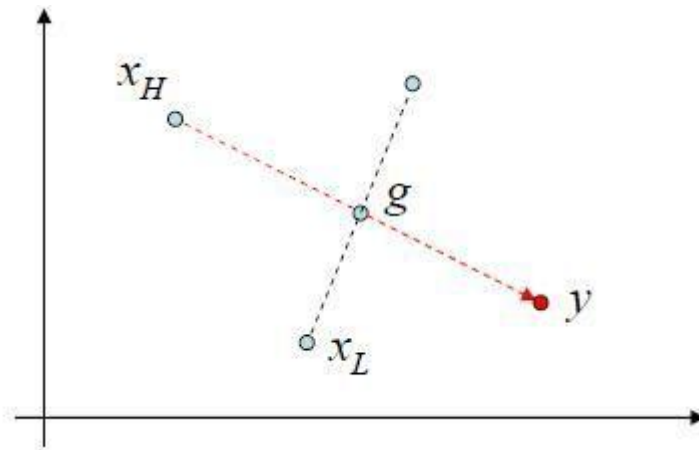
$$x_L = \arg \min_{x \in S} f(x). \quad (6.3)$$

Nyní vypočítáme hodnotu těžiště, jako průměr d bodů po odstranění nejhorší hodnoty x_H .

$$g = \frac{1}{d} \left(\sum_{x \in S} x - x_H \right) \quad (6.4)$$

Nyní hledáme nový bod pomocí reflexe (překlopení x_H přes g), *obr. 20*:

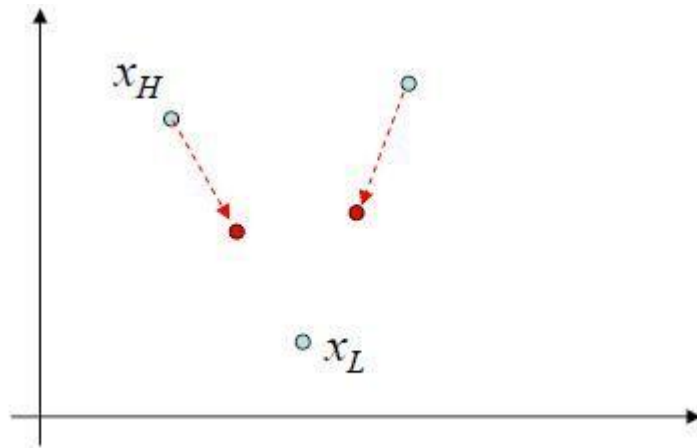
$$y = g + (g - x_H) = 2g - x_H \quad (6.5)$$



Obr. 20 – Reflexe pro $d=2$

Nový bod y musí splnit podmínku $f(y) < f(x_H)$, ten pak nahradí dosud nejhorší bod x_H . Jestliže nedojde ke splnění podmínky, pak místo reflexe použijeme redukci (smrštění simplexu k x_L), *obr. 21*.

$$x \leftarrow \frac{1}{2}(x + x_L), \quad x \in S, x \neq x_L \quad (6.6)$$



Obr. 21 – Redukce pro $d=2$

Reflexe a redukce se opakuje až do splnění ukončovacího kritéria, což je vygenerování simplexu S , jehož vzdálenost je menší než povolená mez 0.01. Z výsledného simplexu vypočítáme pomocí průměru pozici lokálního minima $\text{lok_min} = \text{mean}(\text{simplex})$.

Simplexová metoda byla zvolena po mnoha testech jiných gradientních metod i metod bez znalosti gradientu z Toolbox Matlabu. Byla testována např. gradientní metoda, která je z Toolbox volána příkazem $[x, fval] = \text{fminunc}(\text{fun}, x0)$, kde fun je kriteriální hodnota funkce ve výchozím bodě $x0$. Jako výstup dostáváme pozici lokálního minima x a hodnotu kriteriální funkce $fval$. Tento příkaz je nutné doplnit pomocí `options` o další parametry a podmínky pro správné fungování gradientní metody.

Gradientní metoda fungovala pro postup na kriteriální funkci dobře, avšak při přechodu na tunelovací funkci selhala. Je to dáno tvarem tunelovací funkce, tedy parabolou, kdy krok se postupně zvětšuje velkým rozdílem. Bohužel v této metodě nelze krok sestupu po funkci nastavit. Výsledkem bylo to, že gradientní metoda vyjíměčně některé lokální minimum našla, avšak ve většině případů došlo k přeskočení důležitých minim, někdy dokonce všech.

Ke stejné situaci došlo také u další známé metody $[x, fval] = \text{fminsearch}(\text{fun}, x0)$. Jedná se o modifikaci Simplexové metody, avšak vzhledem k opětovné nemožnosti nastavení kroku či jeho hranic se také ukázala pro tuto problematiku nevhodnou. Proto byla základní Simplexová metoda naprogramována.

K výpočtu hodnoty kritériální funkce, kterou používáme, využijeme SSD metodu (viz rovn. 1.1). SSD metoda je založena na rozdílu intenzit jednotlivých bodů obrazů a je vhodná u obrazů nebo oblastí, které jsou si velmi podobné.

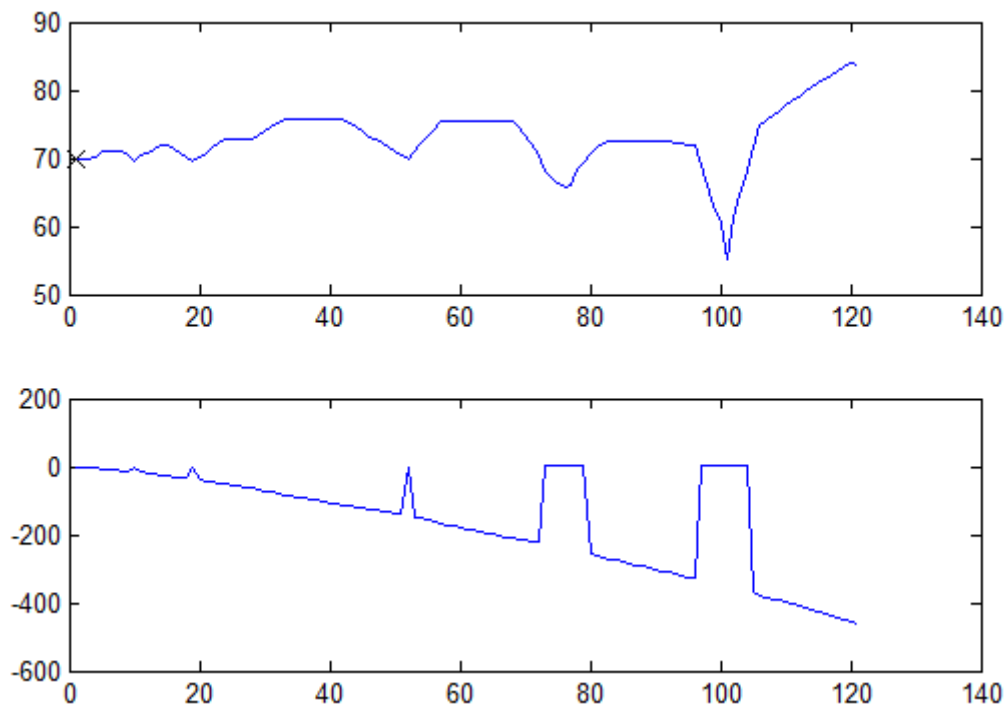
Tunelování pomocí $E(x, x^*)$

Pro vytvoření druhé fáze, tedy tunelování, budeme postupovat dle rovnice pro virtuální funkci $E(x, x^*)$, (rovn. 3.1, popř. 3.2, 3.3). V Matlabu jako:

```
1 function [E] = VirtualniFce(obr1, obr2, lok_min, parametr)
2
3 % keyboard
4 ro = 1;
5 a = 1;
6
7 fxx = TransPodobn(obr1, obr2, lok_min); % fx s hvězdičkou
8 fx = TransPodobn(obr1, obr2, parametr);
9 Esub = log10(1/(1+exp(a-(fx-fxx))));
10 Erep = -(3/4)*ro*((parametr - lok_min)^(4/3))*heaviside(fx-fxx);
11
12
13 E = Esub + Erep;
14 E = real(E);
15
```

Obr. 22 – Virtuální funkce $E(x, x^*)$

Virtuální funkce má maximum v posledním nalezeném lokálním minimu původní funkce $f(x)$. Toto maximum virtuální funkce a minimum původní funkce je položeno rovno nule. Po virtuální funkci klesáme Simplexovou metodou a hledáme lokální minimum. Po nalezení jej promítneme do původní funkce $f(x)$ a dostáváme nový bod, ze kterého opět hledáme nové lokální minimum. Nyní si pro pochopení pojmu virtuální funkce vykreslíme průběh kritériální funkce, který běžně získáme pouze úplným prohledáváním, a virtuální funkci pro první nalezené lokální minimum.



Obr. 23 – Průběh kriteriální funkce nahoře a virtuální funkce dole pro 1. nalezené x^* označené v horním grafu černým křížkem

Z obr. 23 vidíme, že pro hodnoty kriteriální funkce větší než první lokální minimum křivka virtuální funkce klesá ve směru paraboly. V případě nižší hodnoty kriteriální funkce než je hodnota lokálního minima, virtuální funkce skokem roste na nulu. Skoková funkce je zajištěna mimo jiné také Heavisideovou funkcí (rovn. 3.3), což je nespojitá funkce, jejíž hodnota je nulová pro zápornou hodnotu argumentu a rovna jedné pro kladnou hodnotu argumentu. Díky virtuální funkci $E(x, x^*)$ se tedy protunelujeme do nové výchozí pozice (minimum, kde pík roste na 0) na původní funkci $f(x)$. Tato nová pozice je v blízkosti dalšího lokálního (globálního) minima a slouží jako výchozí bod pro opětovné použití Simplexové metody. Díky ní se tedy znovu dostáváme z této výchozí pozice do dalšího lokálního minima, kde se vykreslí nová virtuální funkce náležící příslušnému minimu.

Nalezení nové výchozí pozice, Výstup – GLOBÁLNÍ MINIMUM

V případě, že nalezeneme nové minimum na virtuální funkci, celý cyklus se nám opakuje (tedy nalezení nové výchozí pozice, gradientní metoda pro nalezení lokálního minima a vykreslení nové virtuální funkce). Jestliže ale dojde k opačnému případu, kdy minimum

nalezeno není nebo je překročena hranice prohledávaného prostoru či proběhne maximální povolený počet tunelování, pak celý algoritmus končí. To znamená, že poslední nalezené lokální minimum na původní funkci je považováno za minimum globální. Mimo hodnoty globálního minima také sledujeme počet cyklů a čas, který algoritmus potřebuje pro nalezení globálního minima.

6.2 Realizace TRUST algoritmu pro vícerozměrné případy

Pro realizace algoritmu pro vícerozměrné případy je nutné provést několik úprav na algoritmu pro 1D.

Nejprve je nutné upravit transformační matici, která bude obraz posunovat v ose x a v ose y.

```

1  function par = TransPodobn(obr1,cil,vektor)
2  %vektor(1) = posun po ose X
3  %vektor(2) = posun po ose Y
4
5
6  matice = [1 0 0; 0 1 0; vektor(1) vektor(2) 1];
7
8
9  tform = maketform('affine',matice);
10 cil = imtransform(cil,tform,'XData',[1 size(cil,2)],...
11 'YData',[1 size(cil,1)]);
12
13 [prvni_radky prvni_sloupce] = size(obr1);
14 N = prvni_radky * prvni_sloupce;
15
16 % Odecteni a vypocitani ssd
17 matice_ssd = (obr1 - cil).^2;
18 par = 1/N*sum(sum(matice_ssd));

```

Obr. 24 – Transformační matice pro posuny v x a y, SSD

Dále je nutné zajistit Simplexovou metodu, tedy upravit ji tak, abychom v simplexu měli 3 vstupní body pro plochu, tvořící trojúhelník. Rovnice pro reflexi a redukci nám zůstává a díky této metodě tedy opět nalzáme lokální minimum.

Hlavní je úprava virtuální funkce tak, aby všechny parametry (u nás posuny v ose x a v ose y) byly akceptovány. Použijeme tedy *rovn. 3.4*, kde nám oproti 1D přibyla ve vzorci suma pro 1 až n (v našem případě 1 až 2, tedy pro posun v ose x a v ose y). Suma znázorňuje sumu rozdílů jednotlivých parametrů.

Jako výstup celého algoritmu pro 2D dostáváme hodnoty parametrů x a y , funkční hodnotu globálního minima, počet tunelování a čas potřebný pro nalezení globálního minima.

Ukončovací kritérium volíme jako počet tunelování, případně překročení prohledávaného prostoru.

7 Testování algoritmu TRUST

Algoritmus nyní otestujeme na testovacích funkcích a dále na medicínských obrazech získaných z Ústavu biomedicínského inženýrství v Brně. U testovacích funkcí i u obrazů budeme uvažovat chování algoritmu TRUST vůči metodě Simulované žíhání, dále jen SA. SA je metoda založená na opakovaném zahřívání a ochlazování tělesa (*kap. 3.2.2*), jedná se tedy o iterační algoritmus, který si z Toolbox Matlabu voláme pomocí příkazu `[x, fval, exitFlag, output] = simulannealbnd(@(parametr) TransPodobn(obr1, obr2, parametr), parametr0)`. Do funkce vstupují 2 obrazy či také kritériální hodnota těchto dvou obrazů a výchozí bod. Z výstupu nás bude zajímat hodnota a pozice globálního minima, počet iterací a čas potřebný pro nalezení globálního minima.

7.1 Testování algoritmu na testovacích funkcích

Algoritmus je nyní nutné otestovat na testovacích funkcích zmíněných na v *kap. 4.1*.

Jako první otestujeme algoritmus na jednoduché První De Jongově funkci. Poté přejdeme k testování na složitější Druhé De Jongově funkci. Dále zvolíme funkci Ackleyho, která nám zajistí testování algoritmu na extrém, který může při registraci vzniknout, tedy na mnoha lokálních minimech svažujících se do jednoho ostrého globálního minima. Jako poslední provedeme testování na Rastriginově funkci, která nám také zkoumá chování algoritmu na extrém, tedy na oblasti, kde je velké množství lokálních minim a mezi nimi jedno globální.

První De Jongova funkce

Tvar a minimum této funkce jsme si již zobrazili v *kap. 4.1*. Nyní si ukážeme výsledky testování této jednoduché funkce pomocí algoritmu TRUST a SA. V tabulce vidíme počet iterací nutných k nalezení globálního minima, jeho pozici, funkční hodnotu a čas potřebný pro jeho nalezení.

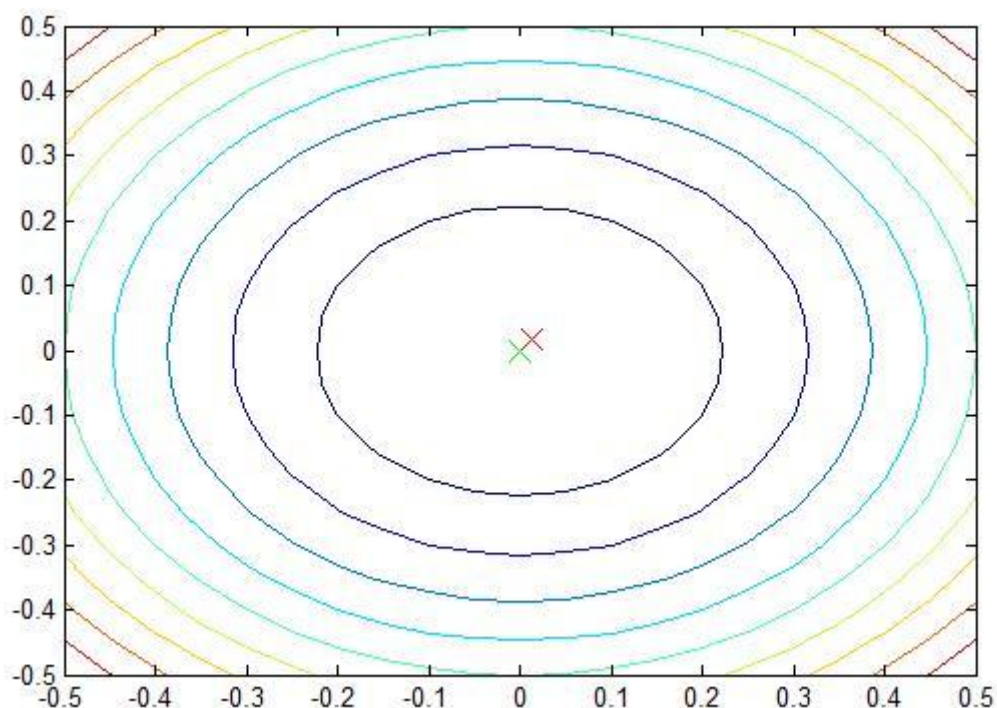
Tabulka 7.1 – Výsledky testování algoritmů na První De Jongově funkci

	počet iterací	pozice globálního minima	funkční hodnota	čas (s)
SA	1000	[0 0]	0	0.7812
TRUST	1	[0,01302 0,01693]	0,0005	0.6150

Výsledkem by mělo být nalezení globálního minima v $x^* = (0)$ a jeho $f(x^*) = (0)$.

Z výsledků v tabulce vidíme, že SA našlo extrém přesně, avšak vytvořený algoritmus TRUST se velmi přiblížil správné pozici minima a v kratším čase.

Výpočetní náročnost je tedy menší v TRUST, avšak za cenu ne zcela přesně definovaného globálního minima. V tomto případě u algoritmu TRUST proběhla pouze první jeho část, tedy nalezení lokálního minima (zde globálního). Nalezení nižších hodnot, tedy funkce tunelování, zde nebyla využita vzhledem k jednoduchosti této testovací funkce, která obsahuje pouze jeden extrém.



Obr. 25 – Srovnání výsledků TRUST a SA na První De Jongově funkci

Na obr. 25 vidíme vykreslení vrstevnic této funkce, kde černý křížek vyznačující správnou pozici globálního minima je překrytý červeným křížkem, který nám značí dobře nalezené minimum SA. Zelený křížek nám zobrazuje TRUST, který se správnému řešení pouze přiblížil avšak v lepším čase.

Pro přehlednost a viditelnost malých rozdílů výsledků jsou vrstevnice vykresleny jen v blízkém okolí globálního minima. Stejně tomu je i u dalších funkcí.

Druhá De Jongova funkce

Výsledek testování na Druhé De Jongově funkci je zobrazen v *tab. 7.2*:

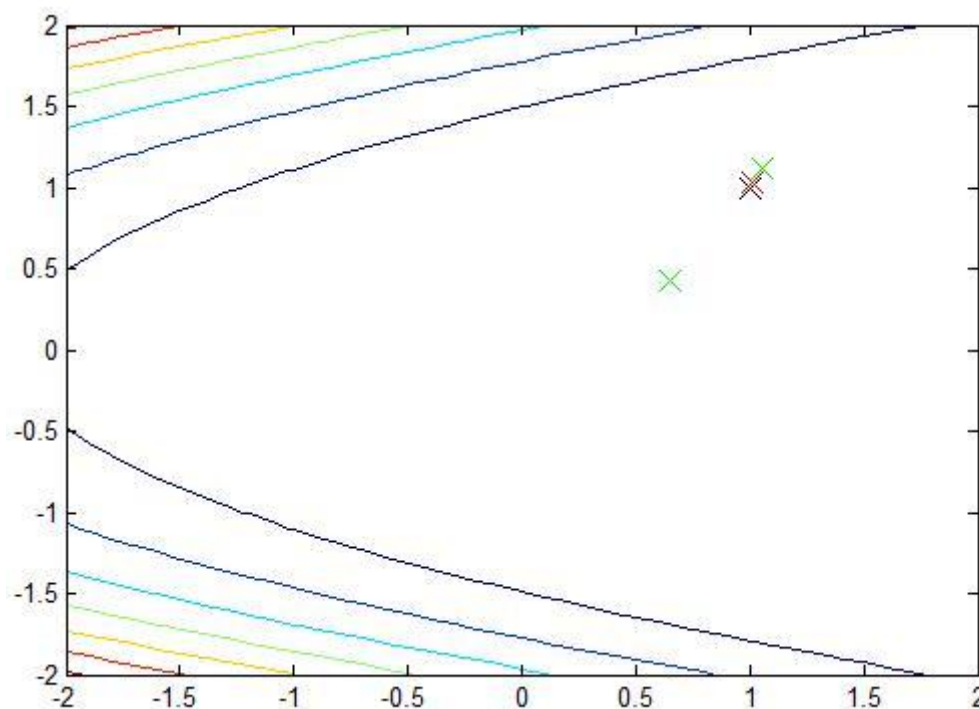
Tabulka 7.2 – Výsledky testování algoritmů na Druhé De Jongově funkci

	počet iterací	pozice globálního minima	funkční hodnota	čas
SA	2806	[1.0128 1.0257]	0	1.5864
TRUST (3)	3	[0.6548 0.4240]	0.1215	1.2644
TRUST (20)	20	[1.0516 1.1113]	0.0055	3.3488

Správná hodnota globálního minima je na pozici $x^* = (1,1)$ s funkční hodnotou v minimu $f(x^*) = 0$.

Zde vidíme opět úspěch SA, kdy minimum je nalezeno na desetiny přesně. V této funkci má větší obtíž nalezení extrému TRUST, který od globálního minima má již větší odchylky než v předchozí funkci.

Algoritmus TRUST byl tedy dále testován a byla upravena hodnota maximálního povoleného počtu tunelování. Z povolených 3 tunelování byla hodnota upravena na 20 tunelování. Po této úpravě bylo dosaženo lepších výsledků, avšak za cenu zvýšení celkového potřebného času pro nalezení správného extrému.



Obr. 26 – Srovnání výsledků TRUST a SA na Druhé De Jongově funkci

Na obr. 26 opět vidíme vyznačené globální minimum černým křížkem, SA červeným a TRUST zeleným. Vzdálenější zelený bod nám vyznačuje nalezené hodnoty pro maximální počet povolených tunelování 3 a přesnější řešení vidíme pro povolených 20 tunelování.

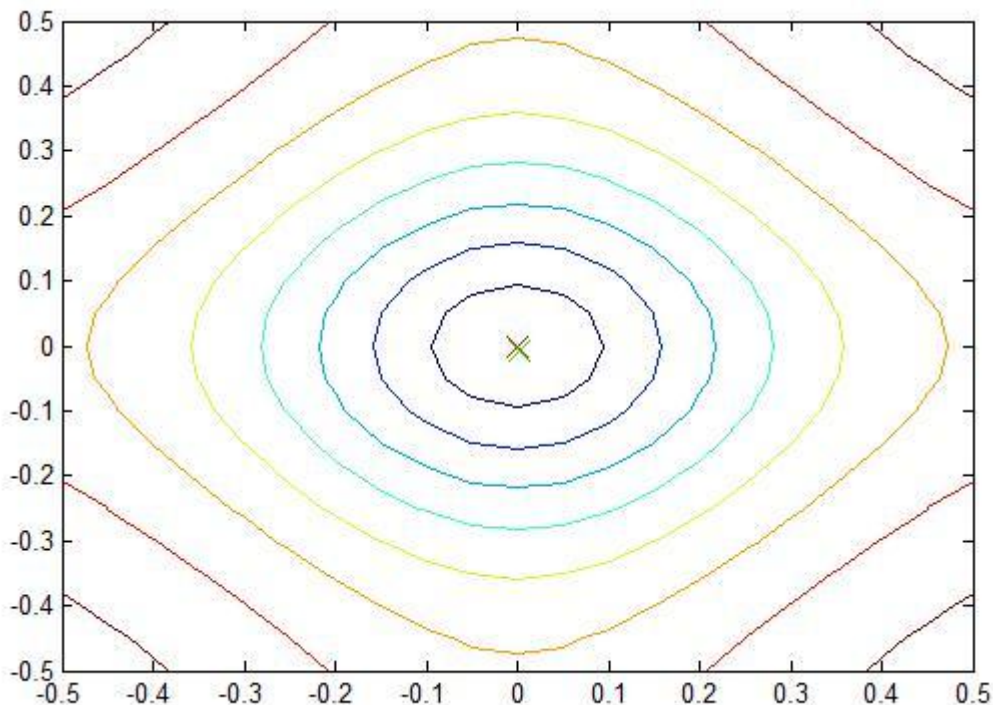
Ackleyho funkce

Další testovanou funkcí je Ackleyho funkce, která má globální minimum v $x^* = (0,0)$ s funkční hodnotou $f(x^*) = 0$.

Tabulka 7.3 – Výsledky testování algoritmů na Ackleyho funkci

	počet iterací	pozice globálního minima	funkční hodnota	čas
SA	1000	[0 0]	2.9805	0.7942
TRUST	3	[0.0012 -0.0088]	2.9827	1.2842

SA našlo extrém zcela přesně a ve velmi dobrém časovém intervalu. TRUST se k minimu velmi přiblížil, avšak jeho čas nebyl lepší než u SA. V grafu opět vidíme zobrazené vrstevnice funkce, kde SA je na stejné pozici jak vyznačené řešení a zelený křížek zobrazující výsledek TRUST, který se ke správnému řešení velmi přiblížil.



Obr. 27 – Srovnání výsledků TRUST a SA na Ackleyho funkci

Tato funkce nám také zobrazuje stav, který může při registraci obrazů vzniknout, tedy mnoho lokálních minim, které se svažují do jednoho ostrého globálního minima. S touto situací si algoritmus TRUST dobře poradil, avšak časovou náročností ani přesností výsledku nepředčil SA.

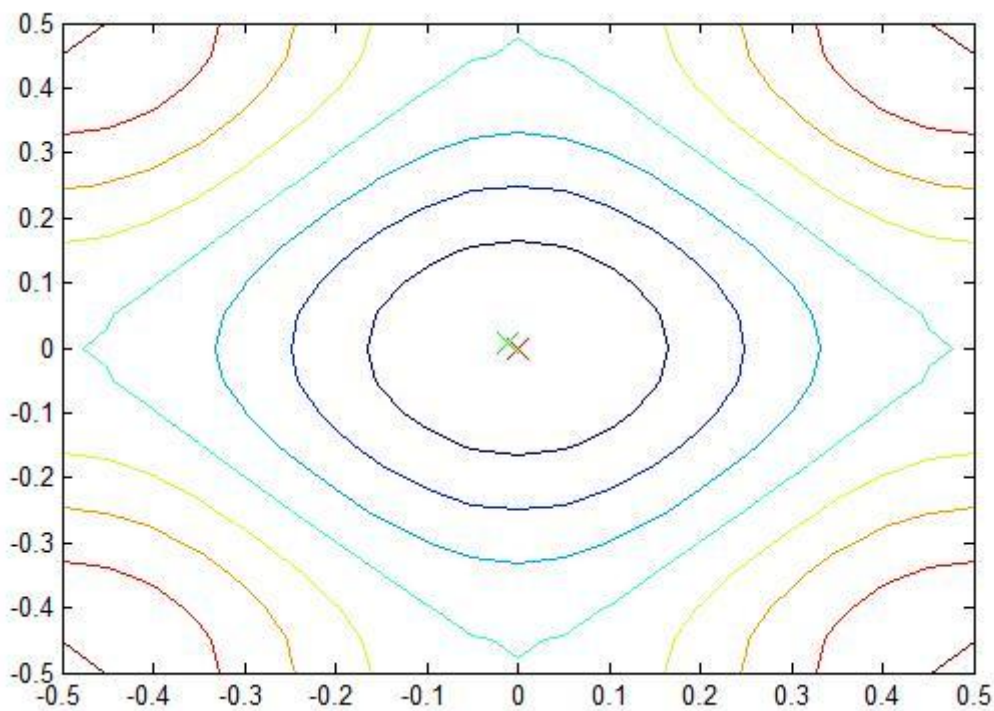
Rastriginova funkce

Poslední testovanou funkcí je Rastriginova funkce, která má opět extrém v $x^* = (0,0)$ s funkční hodnotou $f(x^*) = (0)$.

Tabulka 7.4 – Výsledky testování algoritmů na Rastriginově funkci

	počet iterací	pozice globálního minima	funkční hodnota	čas
SA	1000	[0 0]	0	0.7742
TRUST	3	[-0.0110 0.0078]	0.0361	1.1721

SA opět dosáhlo přesného řešení v dobrém časovém intervalu tak, jako tomu bylo v předchozí funkci. Opět byl testován případ, který může při registraci obrazů vzniknout, tedy velké množství lokálních minim s jedním globálním. TRUST si i s tímto případem poradil, avšak výsledky SA jsou opět lepší.

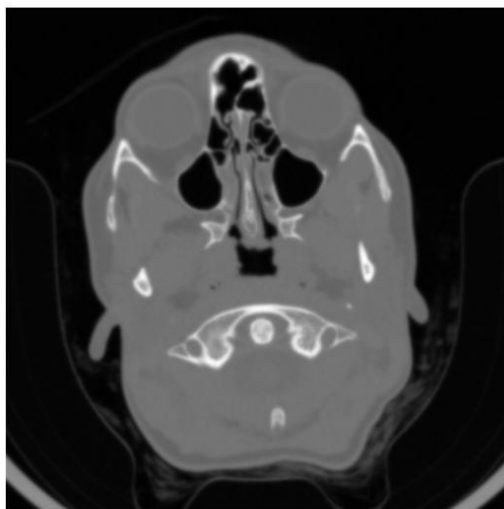


Obr. 28 – Srovnání výsledků TRUST a SA na Rastriginově funkci

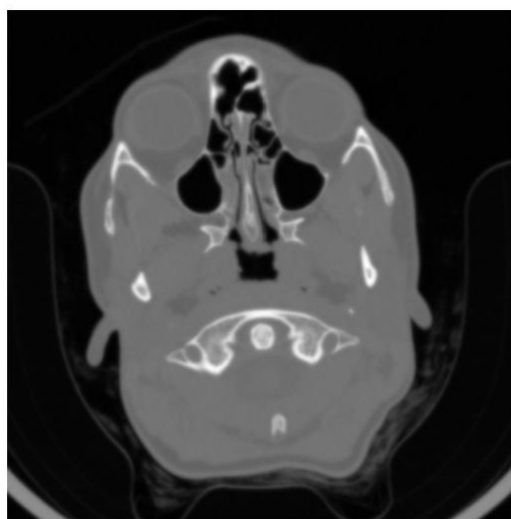
7.2 Testování algoritmu na medicínských obrazech

7.2.1 Testování TRUST na obrazech z CT

Algoritmus nyní budeme testovat na obrazech, které jsou vůči sobě posunuty jak na ose x, tak na ose y. Dále budeme sledovat chování algoritmu po aplikaci šumu s gaussovým rozložením. Testování je provedeno za stejných podmínek na počítači HP 550, procesor Intel Celeron M, paměť 1GB. Prohledávání je provedeno v oblastech posunu ± 10 pixelů.



Obr. 29 – Obraz referenční



Obr. 30 – Obraz pohyblivý

Výsledky testování algoritmů SA a TRUST na medicínských datech z CT máme uvedeny v *tab. 7.5*. Pozice globálního minima je v $x^* = (5,2)$, kde $f(x^*) = 5.7546$.

Tabulka 7.5 – Výsledky testování algoritmů na medicínských datech z CT

	počet iterací	pozice globálního minima	funkční hodnota	čas
SA	2454	[5.0002 2.0014]	5.7546	455.1456
TRUST	2	[4.9748 1.9577]	5.7660	142. 8238

Z výsledků v *tab. 7.5* vidíme, že oba algoritmy se dostatečně přiblížili pozici globálního minima. O něco málo lepších výsledků dosáhlo SA, avšak za cenu trojnásobného času. Otázkou zůstává, jak velkou odchylku od správného řešení lze tolerovat, aby algoritmus našel řešení v přijatelném čase.

Zde pozorujeme, že TRUST dosáhl blízkosti správného řešení, aniž by dosáhl maximálního počtu povolených tunelování v dobrém čase. SA globální minimum našlo za mnohem delší časový úsek, což je dáno výpočetní složitostí či náročností algoritmu, což vidíme z celkového počtu iterací.

Nyní si otestujeme vlastnosti algoritmu v závislosti na šumu. Opět použijeme stejný referenční obraz a na pohyblivý obraz aplikujeme šum s Gaussovským rozložením. To je provedeno pomocí příkazu `obr2= imnoise(obr1, 'gaussian', 0, 0.01)`, kde měníme hodnotu rozptylu šumu 0.001 – 0.01. Výsledky testování jsou uvedeny v *tab. 7.6*:



Obr. 31 – Obraz zatížený šumem s Gaussovým rozložením, rozptyl 0.01

Tabulka 7.6 – Výsledky testování algoritmů po aplikaci šumu s Gaussovým rozložením

	šum	počet iterací	pozice globálního minima	funkční hodnota	čas
SA	0	2454	[5.0002 2.0014]	5.7546	455.1456
TRUST	0	2	[4.9748 1.9577]	5.766	142.8238
SA	0.001	1796	[4.6698 1.6279]	12.9484	690.1939
TRUST	0.001	2	[4.6810 1.6532]	12.9622	137.4841
SA	0.002	2607	[4.6046 1.5734]	18.5739	1005.5623
TRUST	0.002	2	[4.6223 2.4077]	19.4364	116.2482
SA	0.003	1884	[4.5612 1.5557]	23.8149	728.4021
TRUST	0.003	2	[4.5880 1.5469]	23.8256	133.8025
SA	0.004	1772	[4.5551 2.4472]	29.8618	684.5452
TRUST	0.004	2	[4.5497 1.5498]	29.1068	124.7901
SA	0.005	2286	[4.3059 1.5547]	35.7509	886.2503
TRUST	0.005	2	[4.5599 1.5470]	33.8616	118.5167
SA	0.006	1229	[5.4493 1.5316]	39.1718	469.6595
TRUST	0.006	2	[4.5446 1.5288]	38.6727	131.8994
SA	0.007	1413	[4.5422 1.5325]	42.8831	555.8477
TRUST	0.007	2	[4.5417 1.5352]	42.8861	124.3674
SA	0.008	1406	[4.5487 1.5273]	46.9787	585.8635
TRUST	0.008	2	[4.5629 1.5326]	46.9822	142.0558
SA	0.009	2056	[4.5523 1.5536]	50.5979	821.6674
TRUST	0.009	2	[4.5571 1.5483]	50.6069	122.7644
SA	0.01	1625	[4.5527 1.5425]	54.2093	635.8271
TRUST	0.01	2	[4.5493 1.5183]	54.2291	123.2918

Pro upřesnění výsledků si ještě uvedeme výsledné průměry pro počet iterací, pozice globálního minima, funkční hodnoty a čas pro rozptyl šumu 0.001 – 0.01. Dále si uvedeme odchylku vypočtených průměrů parametrů globálního minima a funkční hodnoty od skutečných hodnot, tedy od $x^* = (5,2)$ a $f(x^*) = 5.7546$.

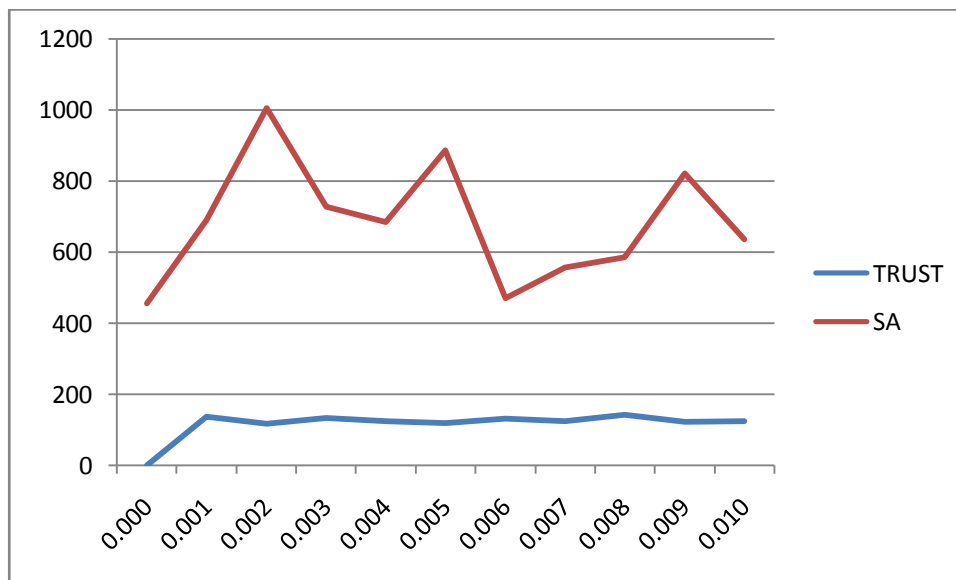
Tabulka 7.7 – Průměr výsledných hodnot a odchylka průměru od skutečných hodnot

	šum	počet iterací	pozice globálního minima	funkční hodnota	čas
Průměr SA	0.001 - 01	1808	[4.6341 1.6446]	35.4791	706.3819
Průměr TRUST	0.001 - 01	2	[4.5757 1.6368]	35.2569	141.8044
Odchylka SA			[0.3659 0.3554]	29.7245	
Odchylka TRUST			[0.4243 0.3632]	29.5017	

Z výsledků uvedených v *tab. 7.6* a *tab. 7.7* můžeme konstatovat, že rychlost algoritmu TRUST zcela zastínila SA. Pokud budeme porovnávat průměr, tak TRUST je pětikrát rychlejší při testování na medicínských obrazech zatížených šumem s Gaussovým rozložením než je tomu u SA. Z funkční hodnoty vidíme, že SA bylo o desetiny přesnější, avšak za cenu mnohem vyššího času. Opět tedy řešíme otázku, zda je pro nás důležitější přesnější řešení či přijatelné řešení ve velmi dobrém čase.

Dále z *tab. 7.6* vidíme, že čím větší je velikost šumu, tím více se vzdalujeme od optimálního řešení. Je tomu jak u TRUST tak u SA.

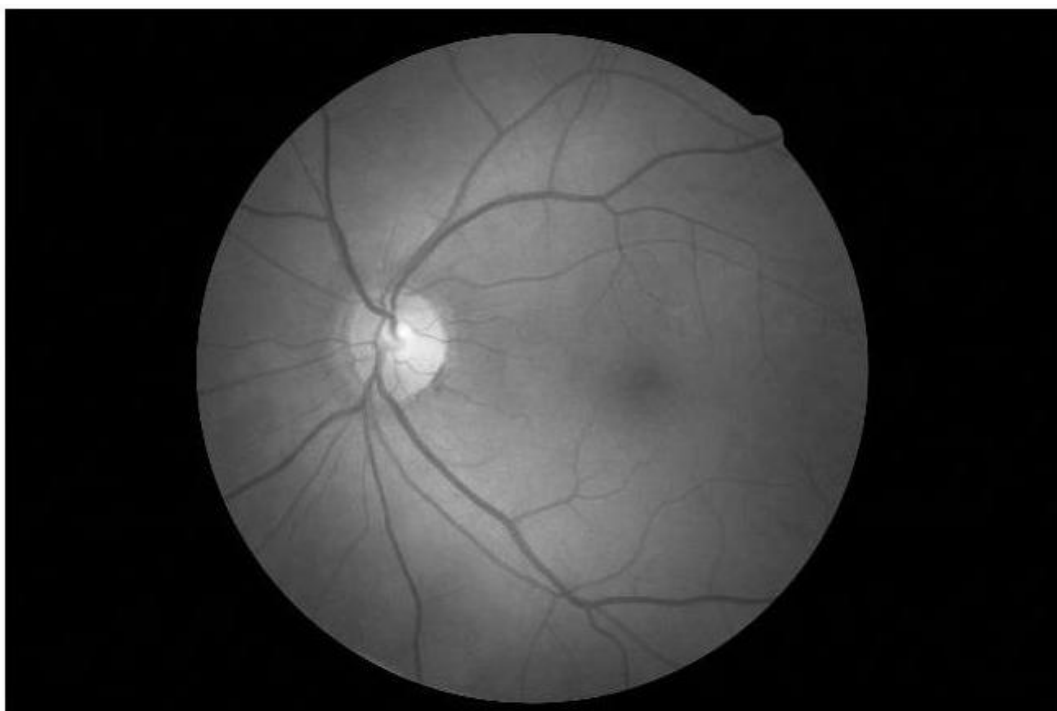
Na *obr. 32* máme zobrazenou závislost času potřebného pro nalezení globálního extrému na velikosti šumu v obraze. Neplatí tedy předpoklad, že čím více šumu, tím delší odezva.



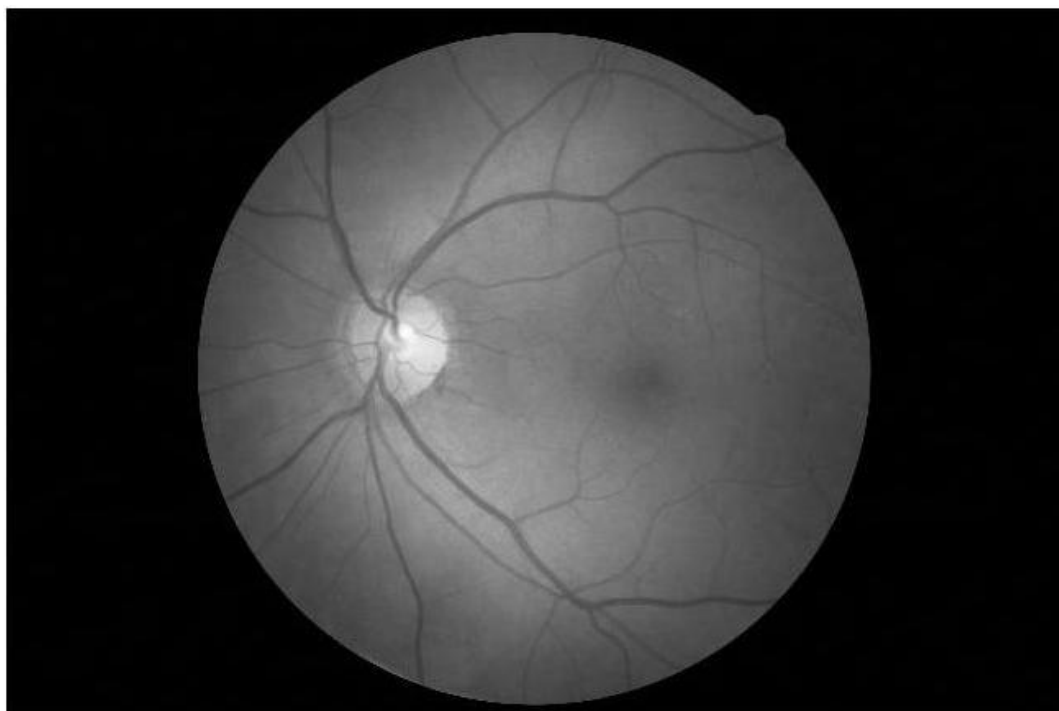
Obr. 32 – Závislost času potřebného pro nalezení extrému na velikosti šumu.

7.2.1 Testování TRUST na obrazech z fundus kamery

Nyní provedeme testování na obrazech z jiné modality, tedy fundus kamery. Jedná se o obrazy sítnice pořízené a získané také z Ústavu biomedicínského inženýrství VUT v Brně. Obrazy mají parametry globálního minima $x^* = (5,3)$, kde $f(x^*) = 3.0360$.



Obr. 33 – Obraz referenční



Obr. 34 – Obraz pohyblivý

Opět provedeme testování algoritmů TRUST a SA, tentokrát však na jiných obrazech, z jiné modality. Výsledky jsou v *tab. 7.8*:

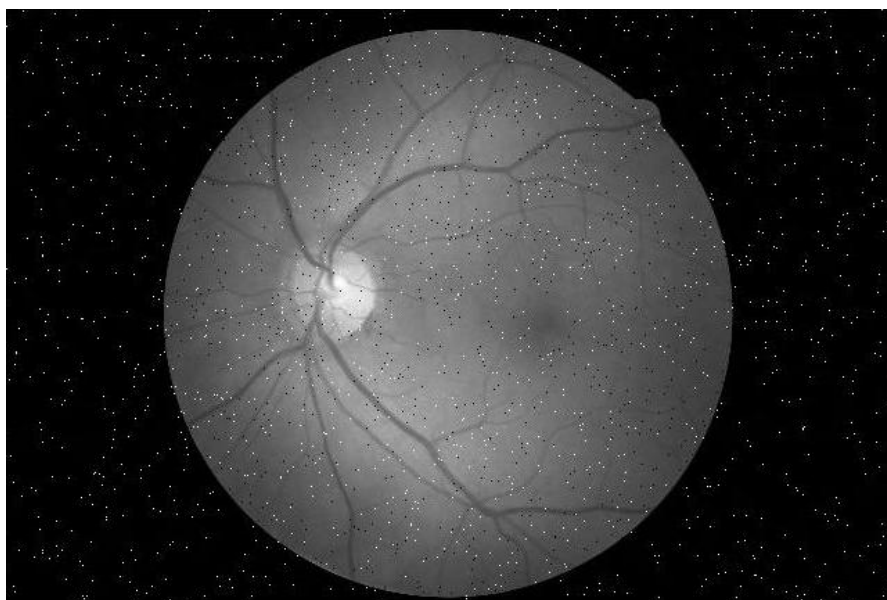
Tabulka 7.8 – Výsledky testování algoritmů na medicínských datech z fundus kamery

	šum	počet iterací	pozice globálního minima	funkční hodnota	čas
SA	0	1184	[4.9991 3.0005]	3.0361	585.1799
TRUST	0	3	[5,0226 3,0029]	3,1224	170.5135

Výsledky jsou obdobné jak u testování algoritmů na obrazech mozku z CT. O setiny vyniká přesností SA, časově je opět lepší algoritmus TRUST.

Velmi důležité je zvolit správný počet tunelování. V případě, že bychom zde povolili větší počet tunelování, pak bychom se možná dostali o setiny k lepšímu výsledku, avšak dalším tunelováním by se nám zvýšila časová náročnost algoritmu, což už by pro nás mohlo být nezajímavé oproti jiným metodám optimalizace.

Vzhledem k tomu, že dosažené výsledky jsou obdobné jak u předchozích obrazů, budeme obrázky, namísto šumu s Gaussovým rozložením, testovat při zatížení obrazu šumem Pepř a sůl. Jedná se o šum, který vzniká během tvorby obrazových dat ztrátou pixelů a vytvoříme jej pomocí příkazu `obr2= imnoise(obr1, 'salt & pepper',0,0.01)`.



Obr. 35 – Obraz znehodnocený šumem Pepř a sůl, hustota 0.01

Bylo provedeno testování na referenčním a pohyblivém obraze, který byl znehodnocen šumem Pepř a sůl s hustotou 0.001 - 0.01.

Tabulka 7.9 – Výsledky testování algoritmů po aplikaci šumu Pepř a sůl

	šum	počet iterací	pozice globálního minima	funkční hodnota	čas
SA	0	1184	[4.9991 3.0005]	3.0361	585.1799
TRUST	0	3	[5,0226 3,0029]	3,1224	170.5135
SA	0.001	1207	[4.9985 2.9998]	3.1113	476.9517
TRUST	0.001	3	[5.0131 3.0027]	3.1373	178.1323
SA	0.002	1203	[5.0040 2.9994]	3.2162	483.9028
TRUST	0.002	3	[5.0245 3.0044]	3.3192	173.6764
SA	0.003	1203	[5.0003 3.0003]	3.2658	475.4865
TRUST	0.003	3	[5.0245 3.0044]	3.3720	174.9176
SA	0.004	1199	[5.0003 3.0013]	3.3521	483.1099
TRUST	0.004	3	[5.0234 3.0036]	3.4657	179.9777
SA	0.005	1198	[4.5551 2.4472]	3.4686	684.5452
TRUST	0.005	3	[5.0122 3.0042]	3.5030	172.5818
SA	0.006	1199	[4.9996 3.0005]	3.5115	479.5559
TRUST	0.006	3	[5.0121 3.0042]	3.5476	168.7798
SA	0.007	1196	[5.0003 2.9990]	3.5991	489.8528
TRUST	0.007	3	[5.0121 3.0042]	3.6371	166.4902
SA	0.008	1221	[5.0004 3.0015]	3.7042	500.5593
TRUST	0.008	3	[4.9949 2.9948]	3.7141	167.9109
SA	0.009	1652	[5.0000 2.9985]	3.7624	671.2429
TRUST	0.009	3	[5.0039 2.9956]	3.7716	171.6669
SA	0.01	1410	[4.9999 2.9982]	3.8655	585.1799
TRUST	0.01	3	[5.0039 2.9956]	3.8755	166.4944

Tabulku doplníme o průměrné hodnoty počtu iterací, parametrů a funkční hodnoty globálního minima a čas potřebný pro jeho nalezení. Dole je opět uvedena odchylka průměru od správných hodnot $x^* = (5,3)$ a $f(x^*) = 3.0360$.

Tabulka 7.10 – Průměr výsledných hodnot a odchylka průměru od skutečných hodnot

	šum	počet iterací	pozice globálního minima	funkční hodnota	čas
Průměr SA	0.001 - 01	1269	[4.9559 2.9446]	3.4857	533.0378
Průměr TRUST	0.001 - 01	3	[5.0123 3.0014]	3.5343	172.0628
Odchylka SA			[0.0441 0.0554]	0.4497	
Odchylka TRUST			[0.0123 0.0014]	0.4983	

Z výsledků v *tab. 7.9* a *tab. 7.10* opět vidíme, že SA je o setiny přesnější, avšak za cenu trojnásobného času potřebného pro nalezení globálního extrému. Funkční hodnota u obou algoritmů s hustotou šumu roste, což znamená, že se pomalu vychyluje od optimálního řešení a čas je na hustotě šumu nezávislý.

Z výsledků vidíme, že algoritmus TRUST je velmi rychlý a zároveň dosahuje přijatelných výsledků. Testován byl na obrazech z dvou různých modalit (CT a fundus kamery), což nemělo na výsledky testování vliv. Dále byly na obrazy aplikovány dva druhy šumů s různou hustotou či rozptylem a to šum s Gaussovským rozložením a šum Pepř a sůl. I s touto situací si TRUST poradil opět ve výborném čase s přijatelným řešením. Ani jeden z šumů nemá vliv na výsledný čas potřebný pro nalezení správného řešení.

Zajímavé by jistě bylo sledovat srovnání SA, TRUST i jiných algoritmů při přidání dalšího rozměru jako je rotace, zkosení či změna měřítko. Dle Benchmarkových testů [30] je u vícerozměrných případů metoda TRUST efektivnější než Genetické algoritmy či Zakázané prohledávání, avšak k takové studii jsem se již nedostala a bylo by zajímavé ji zpracovat v práci větších rozměrů, jako je například disertační.

Na závěr ještě otestujeme předpoklad, že horší přesnost výpočtu pozice a funkční hodnoty globálního minima u algoritmu TRUST může být dána tím, že globální minimum je bráno jako průměr hodnot simplexu (v tomto případě tři hodnot parametrů x a y). Přesnějšího řešení by bylo možné dosáhnout, kdybychom jako řešení brali pozici ze simplexu, která má nejmenší funkční hodnotu. Testování provedeme na obrazech sítnice z fundus kamery, které jsou zobrazeny na *obr. 33* a *obr. 34*.

Tabulka 7.11 – Rozdíl výsledků TRUST

	šum	počet iterací	pozice globálního minima	funkční hodnota	čas
TRUST průměr simplexu	0	3	[5,0226 3,0029]	3,1224	170.5135
TRUST nejmenší funkční hodnota simplexu	0	3	[5,0594 3,0126]	3,5913	212.1843

V *tab. 7.11* v prvním řádku jsou uvedeny výsledky, kdy minimum bereme jako průměr hodnot simplexu. V druhém řádku vidíme výsledky, kdy jako řešení zvolíme pozici s nejmenší funkční hodnotou. K správným hodnotám $x^* = (5,3)$ a $f(x^*) = 3.0360$ se více přibližuje původní řešení, kdy využíváme průměru simplexu. Předpoklad se tedy nepotvrdil, čímž jsme potvrdili správnost celého testování algoritmu TRUST.

Závěr

Optimalizace je hlavní součástí registrace obrazu. V oblasti medicínských obrazů jsou hojně využívány globální optimalizační metody. Tyto byly pro vypracování této práce nezbytné důkladně prozkoumat. Poté byla v první části této práce vypracována jejich řešení.

Jednou z globálních optimalizačních metod je také metoda TRUST, která se skládá ze dvou částí – fáze pro nalezení lokálního minima a fáze tunelování. Tyto dvě části se opakují až do splnění ukončovací podmínky. Tunelování funguje pomocí virtuální funkce, která nám umožní dostat se z lokálního minima do blízkosti dalšího, které má nižší funkční hodnotu. Díky tomuto principu a dobře definovanému ukončovacímu kritériu nalezneme globální minimum. To je velmi podstatné pro dobrou registraci, která je v medicíně velmi důležitá a přispívá k přesnější diagnostice.

Na základě prozkoumání optimalizační metody TRUST bylo možné přejít k návrhu a realizaci algoritmu, který je založený na principu této metody. Algoritmus byl nejprve realizován pro 1D, tedy pouze posun po ose x . Pro nalezení lokálního minima, tedy první fáze metody, byla zvolena po mnoha testech Simplexová metoda. Vzhledem k tvaru virtuální funkce (paraboly), po které algoritmus pro nalezení lokálního minima také pracuje, nebylo možné využít již vytvořené funkce z Toolboxu Matlabu. Funkce velmi dobře pracovaly na původní funkci, avšak při přepnutí na funkci virtuální došlo k přeskočení většiny minim. Vzhledem k nemožnosti nastavení kroku nebylo možné metody použít. Zvolena tudíž byla již zmíněná Simplexová metoda. Po mnoha testech algoritmu je možné říci, že algoritmus vždy najde na jednorozměrné funkci správné řešení.

Poté byl algoritmus rozšířen na 2D případy, tedy pro posun v ose x a y . Testován byl na testovacích funkcích jako je První De Jongova funkce, Druhá De Jongova funkce, Ackleyho funkce či Rastriginova funkce. Algoritmus se vždy přiblížil ke správnému řešení, avšak nikdy nedosáhl přesnosti jako srovnávané Simulované žíhání. Ve srovnání na testovacích funkcích si SA dařilo lépe jak přesností výsledků, tak časově. TRUST byl lepší časově pouze na První De Jongově funkci, kde díky jedinému extrému proběhne pouze první fáze metody, tedy nalezení minima, k tunelování již nedojde, protože nejsou nalezeny hodnoty s menší funkční hodnotou. Ackleyho či Rastriginova funkce nám mimo jiné ukazují také extrémy, které mohou při registraci obrazu nastat (mnoho minim svažujících se do jednoho globálního extrému či mnoho lokálních minim s jedním globálním extrémem). I zde se algoritmus TRUST dokázal dostatečně přiblížit správným hodnotám.

Algoritmus TRUST pro 2D byl dále testován na medicínských datech, tedy obrazech z Ústavu biomedicínského inženýrství v Brně získaných ze dvou různých modalit (CT a fundus kamera). Zde TRUST dokázal svoji rychlost, která byla až třikrát větší než u SA.

TRUST dosáhl uspokojivého výsledku maximálním povoleným počtem tunelování. U SA došlo ke zpomalení velkým počtem iterací, tedy byla zde ukázána velká výpočetní náročnost ve složitějších případech. Obrazy byly také znehodnoceny šumem a testovány pro jeho různou hustotu či rozptyl. Aplikován byl šum s Gaussovým rozložením a šum Pepř a sůl. I přes tuto situaci dokázal TRUST najít přijatelné řešení ve výborném čase.

Velmi zajímavé by bylo porovnání pro vícerozměrné případy, ke kterému se bohužel již v této práci nedostalo. Pokračováním práce by mohlo být právě zmíněné rozšíření algoritmu o více rozměrů. Dále by bylo velmi zajímavé zkoumat metody nalezení lokálního minima pro tento algoritmus, protože těchto metod je veliké množství a také ovlivňují přesnost a rychlost algoritmu.

Algoritmus byl tedy vytvořen pro 1D a pro 2D případy. Byl testován jak na velkém počtu vytvořených funkcí, tak na testovacích funkcích. Na závěr byl testován i na medicínských obrazech z různých modalit a na obrazech zatížených různým druhem a hustotou šumu. Z výsledků a srovnání se SA vidíme, že TRUST dosahuje slušných výsledků, obzvláště při srovnání na medicínských datech.

Velice kladně hodnotím přepis původní funkce na funkci virtuální na *obr. 23*. V případě více času by bylo jistě zajímavé s tímto algoritmem nadále pracovat, protože při důkladném doplnění a úpravě by mohl předčít známější a používanější metody jako jsou Genetické algoritmy, Simulované žíhání či Zakázané prohledávání.

Použitá literatura

- [1] AIT-AOUDIA, S.; MAHIOU, R.: *Medical Image Registration by Simulated Annealing and genetic algorithms. IEEE Congress on Geometric Modeling and Imaging.*, 2007, [online]. [cit. 2012-2-20]. Dostupný z: <http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4271735>
- [2] BARHEN, J.: *TRUST: A Deterministic Algorithm for Global Optimization* [online]. [cit. 2012-3-3]. Dostupný z: <<http://www.sciencemag.org/content/276/5315/1094.full>>
- [3] BARTOŠ, M.: *Lícování obrazů lékařských zobrazovacích modalit*, Brno: Diplomová práce, FEKT VUT, 2007, 66 s.
- [4] BISTRÝ, J.: *Registrace obrazu pomocí metody optical flow*, [online]. Brno: Diplomová práce, FEKT VUT. [cit. 2012-2-18]. Dostupný z: <https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=37902>
- [5] BROWN, L. G.: *A survey of image registration techniques.* ACM Comput. Surv. 24, s. 325 - 376
- [6] CESAR – CENTER FOR ENGINEERING SCIENCE ADVANCED RESEARCH.: *TRUST: Advances in Global Optimization* [online]. [cit. 2012-3-3]. Dostupný z: <<http://www.cesar.ornl.gov/complex-index2.html>>
- [7] CETIN, B. C.; BARHEN, J.; BURDICK, J. W.: *Terminal Repeller Unconstrained Subenergy Tunneling (TRUST) for fast Global Optimization*, Journal of Optimization Theory and Applications, vol. 77, no. 1, pp. 97-126,1993
- [8] CETIN, B. C.; BURDICK, J. W.; BARHEN J.: *Global descent replaces gradient descent to avoid local minima problem in leasing with artificial neural network* [online]. [cit. 2012-3-3]. Dostupný z: <http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=298667>
- [9] ČAPEK, M.: *Registrace snímků z konfokálního mikroskopu.* ČVUT FEL, Praha, 1999
- [10] FANELLI, S.: *A New Algorithm for Box-Constrained Global Optimization* [online]. [cit. 2012-4-1]. Dostupný z: <<http://www.springerlink.com/content/ln85u1u26n26587w/>>
- [11] FRISTON, K. J.; PENNY, W. D.: *Statistical Parametric Mapping, The Analysis of Functional Brain Images.* London, 2007
- [12] GENETICKÉ ALGORITMY: *Genetické algoritmy* [online]. [cit. 2011-10-18]. Dostupný z: <[http:// poli.cs.vsb.cz/edu/isy/down/ga.pdf](http://poli.cs.vsb.cz/edu/isy/down/ga.pdf)>
- [13] GLOVER, F.: *Tabu search – Part 1. ORSA Journal on computing.*, s. 190 – 206, 1989

- [14] HEURISTIC LAB: *Real Valued Test Functions* [online]. [cit. 2012-3-3]. Dostupný z: <<http://dev.heuristiclab.com/trac/hl/core/wiki/TestFunctions>>
- [15] HOLDEN, M.; HILL, D. L. G.; DENTON, E. R. E.: *Voxel similarity measures for 3-D serial MR brain image registration. IEEE Congress on Medical Imaging.*, 2000, [online]. [cit. 2012-2-26]. Dostupný z: <http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=836369>
- [16] CHEN, Y.; BROOKS, R. R.; IYENGAR, S. S.; RAO, N. S. V.; BARHEN, J.: *Efficient global optimization for image registration, Knowledge and Data Engineering, IEEE Transactions on* [online]. [cit. 2011-11-07]. Dostupný z: <<http://www.cesar.ornl.gov/publications/pub-pdfs/Rao-Eff58A.pdf>>
- [17] KOZUMPLÍK, J.: *Evoluční algoritmy* [online]. [cit. 2012-2-18]. Dostupný z: <<https://www.vutbr.cz/elearning/mod/resource/view.php?id=141014>>
- [18] KOZUMPLÍK, J.: *Evoluční algoritmy a neuronové sítě: Chapter_09* [online]. [cit. 2011-10-25]. Dostupný z: <http://www.kiv.zcu.cz/studies/predmety/uir/texty/Chapter_09.pdf>
- [19] LUNER, P.: *Jemný úvod do genetických algoritmů* [online]. [cit. 2011-10-18]. Dostupný z: <<http://cgg.mff.cuni.cz/~pepca/prg022/luner.html>>
- [20] MÉZL, M.: *FEAL: Genetický algoritmus* [online]. [cit. 2011-10-18]. Dostupný z: <<https://www.vutbr.cz/elearning/mod/resource/view.php?id=147409>>
- [21] MODERNÍ METODY OPTIMALIZACE: *TABU search metoda* [online]. [cit. 2011-10-18]. Dostupný z: <http://klobouk.fsv.cvut.cz/~leps/teaching/mmo/prednasky/prednaska05_Tabu.pdf>
- [22] MOLGA, M.; SMUTNICKI, C.: *Test function for optimization needs* [online]. [cit. 2012-3-3]. Dostupný z: <<http://www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf>>
- [23] OBLOW, E. M.: *SPT: A stochastic tunneling algorithm for global optimization* [online]. [cit. 2012-4-1]. Dostupný z: <<http://www.springerlink.com/content/mm070q3p46211tg0/fulltext.pdf>>
- [24] PANUŠ, J.: *Evoluční algoritmy v optimalizačních problémech veřejné správy.*, s. 20 – 22, Pardubice, 2008
- [25] RAHNAMAYAN, S.; TIZHOOSH, H. R.; SALAMA, M. M. A.: *Opposition-Based Differential Evolution* [online]. [cit. 2012-3-3]. Dostupný z: <http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4358759>
- [26] RHODY, H.: *Geometric image transformations* [online]. [cit. 2012-2-18]. Dostupný z: <http://www.cis.rit.edu/class/simg782/lectures/lecture_18/lec782_04_17.pd>

- [27] ROBINSON, J.; RAHMAT – SAMII, Y.: *Particle swarm optimization in electromagnetics*, *IEEE Trans Antenas Propagat* 52, s. 397 – 407, 2004
- [28] ROBINSON, J.; SINTON, S.; RAHMAT – SAMII, Y.: *Particle swarm, genetic algorithm, and they hybrids: optimazition of a profiled corrugated horn*. *Antennas and Propagation Society International Symposium*, 2002, [online]. [cit. 2012-2-26]. Dostupný z: <http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1016311>
- [29] SCHWARZ, D.; PROVAZNÍK, I.: *Registrace multimodálních medicínských obrazů s využitím matlabu* [online]. [cit. 2011-11-07]. Dostupný z: <http://dsp.vscht.cz/konference_matlab/matlab04/schvarz.pdf>
- [30] SRINIVAS, M.; RANGAIAH, G. P.: *Implementation and evaluation of random tunneling algorithm for chemici engineering applications* [online]. [cit. 2012-4-1]. Dostupný z: <<http://www.sciencedirect.com/science/article/pii/S0098135406000706>>
- [31] SURI, J. S.; WILSON, D. L.; LAXMINARAYAN, S.: *Handbook of Biomedical Image Analysis* [online]. [cit. 2012-2-26]. Dostupný z: <<http://www.scribd.com/doc/57992233/Kluwer-Handbook-of-Bio-Medical-Image-Analysis-Vol-3>>
- [32] TING, C. K.; LI, S. T.; LEE, C.: *A new integrated approach to evolutionary algorithms*. *IEEE Congress on Evolutionary Computation.*, 2001, [online]. [cit. 2011-10-18]. Dostupný z: <<http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7440>>
- [33] TVDÍK, J.: *Evoluční algoritmy.*, s. 35-39, Ostrava, 2004
- [34] WEISE, T.: *Global Optimization Algorithms – Theory and Application*, 2009 published by it-weise.de (self-published), Germany. 2009
- [35] YUANJING, F.; ZUREN, F.; LI, Y.: *Ant Colony Optimization for Phase Change Image Sequences Segementations*. *IEEE Congress on Intelligent Control and Automation.*, 2006, [online]. [cit. 2012-2-20]. Dostupný z: <http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1713181>
- [36] ZHANG, H.; ZHOU, X.; SUN, J.; ZHANG, J.: *A novel medical image registration Metod based on mutual information and genetic algorithm.*, s. 221-226, 2005
- [37] ZITOVÁ, B.: *Image Registration methods: a survey* [online]. [cit. 2012-2-18]. Dostupný z: <<http://library.utia.cas.cz/prace/20030125.pdf>>