

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE  
Fakulta životního prostředí  
Katedra vodního hospodářství a enviromentálního modelování  
Praha

**Analýza globální optimalizační metody PSO**

Bakalářská práce

**autor: Martina Dubská**  
**vedoucí práce: doc. Ing. Petr Máca, Ph.D.**

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně, pod vedením doc. Ing. Petra Máci, Ph.D., a že jsem uvedla všechny literární prameny, ze kterých jsem čerpala.

V Praze 14. dubna 2015

Martina Dubská

# ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Katedra vodního hospodářství a environmentálního modelování

Fakulta životního prostředí

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Martina Dubská

Vodní hospodářství

Název práce

**Analýza globální optimalizační metody PSO**

Název anglicky

**Analysis of global optimisation method PSO**

---

### Cíle práce

Cílem práce je analýza globální optimalizační metody PSO. Práce by měla poskytnout základní rozbor algoritmu PSO, variant PSO a jeho aplikací v hydrologickém modelování.

### Metodika

Při řešení práce se zaměřte na

1. Proveďte popis optimalizací a rozdělení optimalizačních algoritmů. Vypracujte souhrn testovacích optimalizačních úloh.
2. Popište základní algoritmus PSO.
3. Uveďte vybrané modifikace PSO.
4. Proveďte rozbor vybrané implementace PSO v hydrologii.

**Doporučený rozsah práce**

rozsah 30 -50 stran

**Klíčová slova**

optimalizace, globální optimaliza, optimalizace rojem částic

---

**Doporučené zdroje informací**

fulltexty ze standardních vědeckých periodik

Kennedy, J. – Eberhart, R. C. – Shi, Y. Swarm intelligence. Kaufmann, San Francisco, 2001,

Zambrano-Bigiarini, M. – Rojas, R. A model-independent Particle Swarm Optimisation software for model calibration. Environmental Modelling & Software. 2013, 43,

Zelinka, I. et al. Evoluční výpočetní techniky- principy a aplikace. BENtechnická literatura, 2009. ISBN 978-80-7300-218-3.

---

**Předběžný termín obhajoby**

2015/06 (červen)

**Vedoucí práce**

Ing. Petr Máca, Ph.D.

---

Elektronicky schváleno dne 14. 4. 2015

**prof. Ing. Pavel Pech, CSc.**

Vedoucí katedry

---

Elektronicky schváleno dne 14. 4. 2015

**prof. Ing. Petr Sklenička, CSc.**

Děkan

V Praze dne 15. 04. 2015

## **Poděkování**

Ráda bych poděkovala svému vedoucímu práce doc. Ing. Petrovi Mácovi, Ph.D. za ochotu, trpělivost, čas a cenné rady, které mi během psaní mé práce věnoval.

## **Abstrakt**

Cílem této bakalářské práce je analýza globální optimalizační techniky PSO- optimalizace hejnem částic a její využití v hydrologii. V první části se zabývám popisem a rozdělení optimalizací, dále jsou v práci popsány testovací optimalizační úlohy, které slouží k ověřování výkonnosti algoritmů. V další části je popsán samotný algoritmus optimalizace hejnem částic. Následně jsou vybrány některé modifikace algoritmu a jejich popis. V poslední části práce jsou uvedeny příklady použití v hydrologii a popis implementace algoritmu v hydrologii.

## **Klíčová slova**

optimalizace, globální optimalizace, optimalizace rojem částic

## **Abstract**

The aim of this thesis is to analyze the global optimization technique, to be more precise particle swarm optimization and its application in hydrology. The first part deals with description and distribution of optimization, furthermore, the paper describes the benchmark functions, which are use to verify the performance of optimization algorithms. The next section describes particle swarm optimization itself. Subsequently, several modifications of algorithm are selected and described. The thesis concludes with providing the examples of application of PSO in hydrology.

## **Key words**

optimization, global optimization, particle swarm optimization

# OBSAH

<b>Obsah</b>	8
<b>1 Úvod</b>	11
<b>2 Optimalizace</b>	13
2.1 Úvod	13
2.2 Rozdělení optimalizací	14
2.3 Testovací funkce	18
2.3.1 Neomezené multi-modální testovací funkce	18
2.3.2 Neomezené uni-umodální testovací funkce	21
2.3.3 Testovací funkce omezené hranicemi	24
<b>3 Popis PSO</b>	27
3.1 Popis algoritmu	28
3.2 Topologie	33
<b>4 Modifikace PSO</b>	35
4.1 Celulární optimalizace hejnem částic	35
4.2 Centrální optimalizace hejnem částic	36
4.3 Adaptivní optimalizace hejnem částic	36
4.4 Optimalizace Multi-hejnem	36
4.5 Další vybrané modifikace PSO	37
<b>5 PSO v hydrologii</b>	39
5.1 Aplikace na povodí	40
5.2 HydroPSO	40
5.2.1 Popis balíku hydroPSO	41
5.2.1.1 Funkce hydroPSO	42
5.2.1.2 Případové studie	43
<b>6 Diskuse</b>	45
<b>7 Závěr</b>	47







# Kapitola 1

## ÚVOD

Optimalizace je proces, jenž nás má dovést k tomu nejlepšímu možnému řešení konkrétního problému, jde o výběr nejlepší varianty z dané množiny variant. Optimalizace se využívá v různých procesech, odvětvích a vědních disciplínách až po běžné procesy lidské činnosti, např. únikový plán bodovynalezení neoptimalnější cesty nebo tvorba školního rozvrhu. Optimalizační algoritmy jsou velmi významným nástrojem pro řešení problémů inženýrské praxe. Většinou se využívají tam, kde je řešení daného problému analytickou cestou nevhodné či nereálné. (Mañas, 1979)

Optimalizační úlohy mají svůj původ v historii, již v antice byly řešeny úlohy o nalezení největších či nejmenších hodnot. Mezi nejznámější patří Didonina úloha, která je spjata s pověstí o založení Kartága. V 17. století Johann Bernoulli publikoval tzv. úlohu o brachystochroně, kde je úkolem najít tvar křivky spojující body A a B, po které se těleso dostane z bodu A do bodu B za nejkratší dobu. V roce 1687 napsal Isaac Newton knihu "Metematické základy přírodní filozofie" a tím přispěl k rozvoji optimalizační problematiky. Velký rozvoj nastal po druhé světové válce, kdy se optimalizační úlohy věnovaly hlavně ekonomice. (Zelinka et al., 2009).

Vzhledem k tomu, že do tohoto procesu vstupuje mnoho proměnných, je to proces velmi obsáhlý a obtížný. Nedá se vždy použít jednoznačné řešení. Optimalizační proces není nikdy konečný a stoprocentní, nepodává nám pravdu o skutečnosti, ale jen se nám ji snaží co nejvěrněji vykreslit.

V hydrologii je optimalizace užitečná hlavně při předpovědi povrchového odtoku, kvality vody, kalibraci hydrologického modelu, meteorologické předpovědi nebo k posouzení srážkovo-odtokového procesu. PSO byla aplikována v celé řadě hydrologických studií, např. k předpovědi povodně (Chau, 2006), ke kalibraci hydrologického modelu hydroPSO (Zambrano-Bigiarini – Rojas, 2013), ke studii srážko-odtokového procesu.

V méj bakalářské práci se budu zajímat o optimalizaci hejnem částic (dále jen PSO- z anglického překladu particle swarm optimization) a její využití v hydrologii. V méj práci se budu snažit zjistit na kolik se tato optimalizace přibližuje skutečnosti, jakým způsobem je možné zlepšit a o kolik se tyto optimalizace liší. Cílem práce je rešeršní zpracování optimalizace hejnem částic a její využití v hydrologii. Práce je rozdělena na čtyři části. V první

části jsou popsány optimalizace, jsou zde uvedena rozdělení a na konci jsou vybrány některé testovací funkce. Druhá část popisuje algoritmus PSO, dále jsou uvedeny příklady modifikací PSO a nakonec použití PSO v hydrologii.

# Kapitola 2

---

## OPTIMALIZACE

### 2.1 Úvod

Předmětem matematické optimalizace je minimalizovat nebo maximalizovat funkce mnoha proměnných za případných omezujících podmínek. Do této definice se dá zahrnout množství úloh z inženýrské praxe a přírodních věd. Optimalizaci je možné pojmenovat i jako matematické programování, lze ji zařadit do oboru aplikované matematiky, ležící na pomezí matematické analýzy, lineární algebry a informatiky. Problém který má být vyřešen, je nutno převést do matematického modelu ( zvolit proměnné, účelovou funkci a omezení). (Werner, 2011)

Optimalizační úlohy lze formulovat jako hledání minima dané reálné funkce na dané množině:

$$\min_{x \in X} f(x) \quad (2.1)$$

Kde funkce  $f$  je funkce účelová, množina  $X$  je množina přípustných řešení, prvky množiny  $\operatorname{argmin}_{x \in X}$  jsou optimální řešení.

Úlohy vedoucí k výpočtu extrému funkcí více proměnných se často získávají jako matematické formalizace technických nebo ekonomických problémů, v nichž jde o výběr nejlepší varianty z dané množiny variant. Jednotlivé varianty jsou v této formalizaci charakterizovány  $n$ -ticí čísel (bodem  $X = [x_1, x_2, \dots, x_n]$ ) a ohodnocení varianty je dáno hodnotou funkce funkce  $f$  v tomto bodě  $X$ . Bod ve kterém funkce nabývá maxima nebo minima, reprezentuje potom hledanou nejlepší variantu (Mañas, 1979).

Postup hledání minima a maxima funkce je rovnocenný. Pojmy minimalizace, maximalizace a optimalizace jsou lehce zaměnitelné. Optimalizační úlohu definujeme funkcí  $f$  a  $n$ -dimenzionálním prostorem  $R^n$ .

$$f : R^n \rightarrow \mathbb{R} \quad (2.2)$$

Matematický model se však nikdy nerovná skutečnosti, dává nám jen přibližnou představu o situaci.

## 2.2 Rozdělení optimalizací

Optimalizační metody lze rozdělit podle mnoho kritérií. V následující podkapitole je uvedeno pár příkladů jejich rozdělení.

Optimalizační algoritmy vyhledávají minima účelové funkce, tak že hledají optimální numerickou kombinaci jejich argumentů. Tyto algoritmy lze rozdělit podle principů jejich činnosti, podle složitosti algoritmu atp. Algoritmy lze rozdělit do tří skupin: enumerativní, deterministické a stochastické.

- **Stochastické algoritmy**, kam patří i PSO, náhodně prohledávají hodnoty argumentů a výsledkem je nejlepší řešení. Tyto algoritmy nejsou vhodné pro přesný odhad, větší prohledávané území a jsou pomalejší.
- **Deterministické algoritmy** mají pokaždé stejné výsledky, jsou tedy předvídatelné. Výsledek je jedno řešení.
- **Enumerativní algoritmy** vypočítává všechna možná řešení daného problému. Je vhodný pro problémy, kde jsou argumenty účelové funkce diskrétního charakteru a nabývají nízkých hodnot. Pokud by byl použit obecně, zcela reálně by mohl potřebovat na úspěšné ukončení čas, který je delší než existence našeho vesmíru. ([Zelinka et al., 2009](#))

([Weise, 2008](#)) se také zabývá rozdělením optimalizací. Globální optimalizace dělí na deterministické a pravděpodobnostní. Deterministické algoritmy jsou předvídatelné, výsledkem je vždy jen jedno řešení. Pravděpodobnostní algoritmus má výsledek pokaždé jiný. Rozhoduje se náhodně. Dále optimalizace dělí podle rychlosti řešení na online a offline optimalizace. Online optimalizace se pohybují v rozmezí milisekund až minut a offline optimalizace se pohybují v rozmezí několika dnů, tudíž zde čas nehraje roli. Rozdělení je zobrazeno [2.1](#)

Další možné rozdělení uvádí ([Mendes, 2004](#)), kde optimalizační metody rozděluje na exaktní, heuristické a metaheuristické.

### 1. Exaktní

Lineární programování

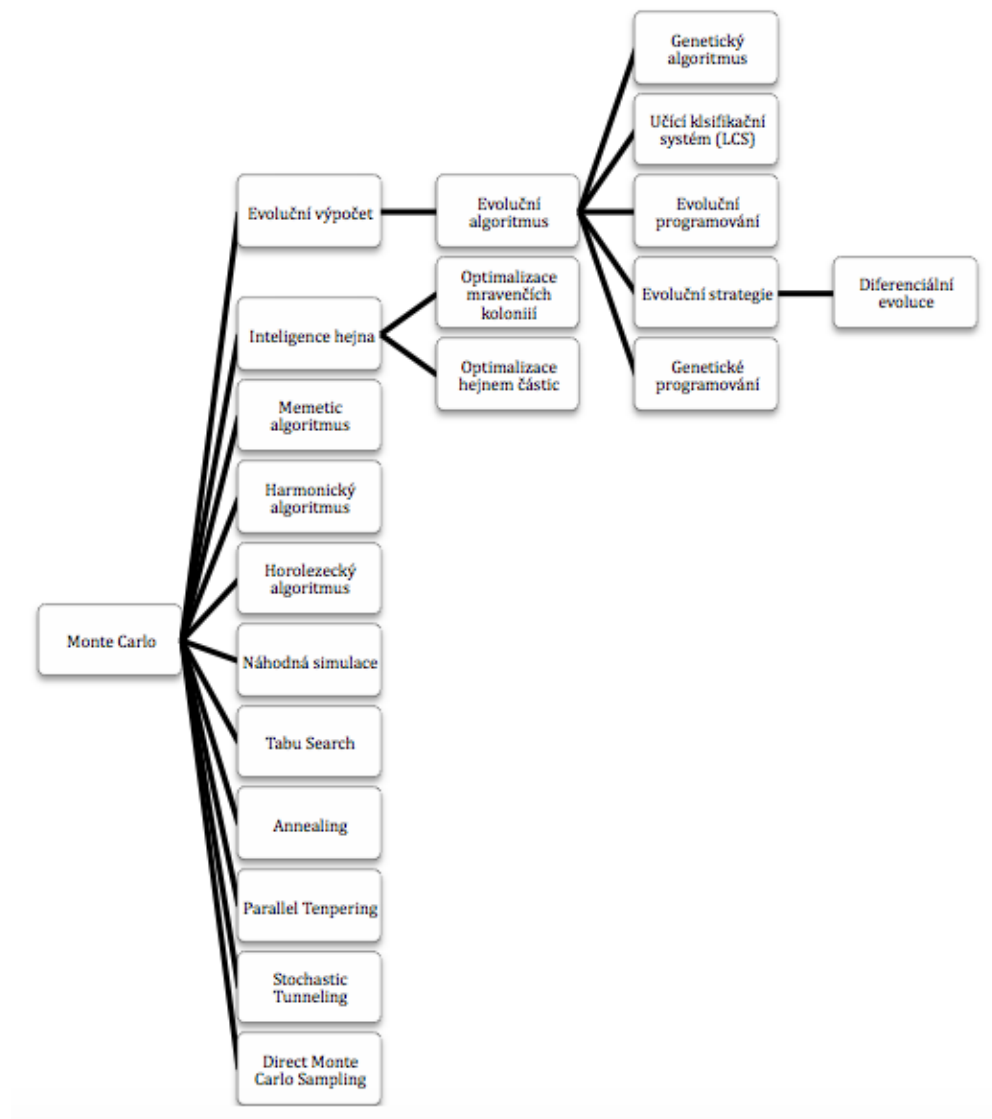
Dynamické programování

Metoda rozděl a panuj

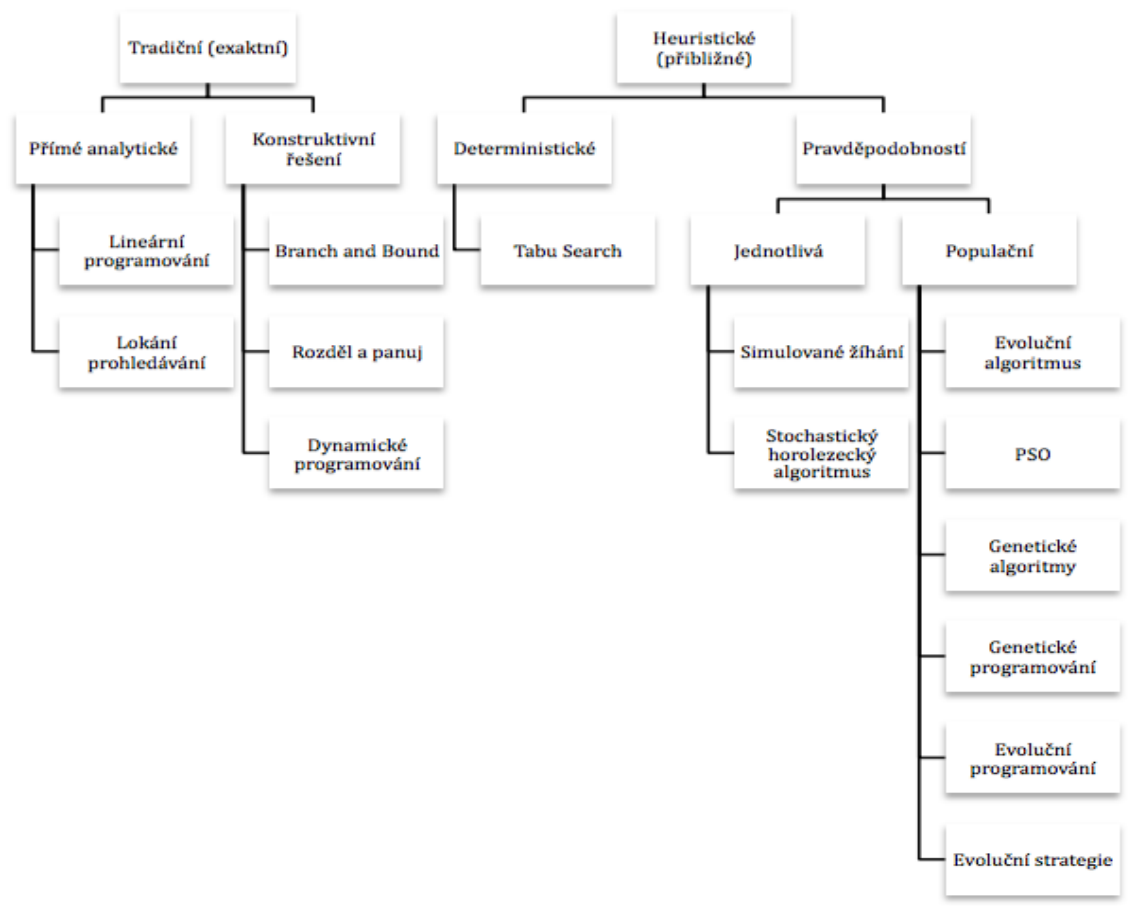
### 2. Heuristické

Metoda pokus omyl

Nelder- Meadův algoritmus



Obrázek 2.1: Rozdělení podle (Weise, 2008)



Obrázek 2.2: Rozdělení podle (Riget – Vesterstrom, 2002)

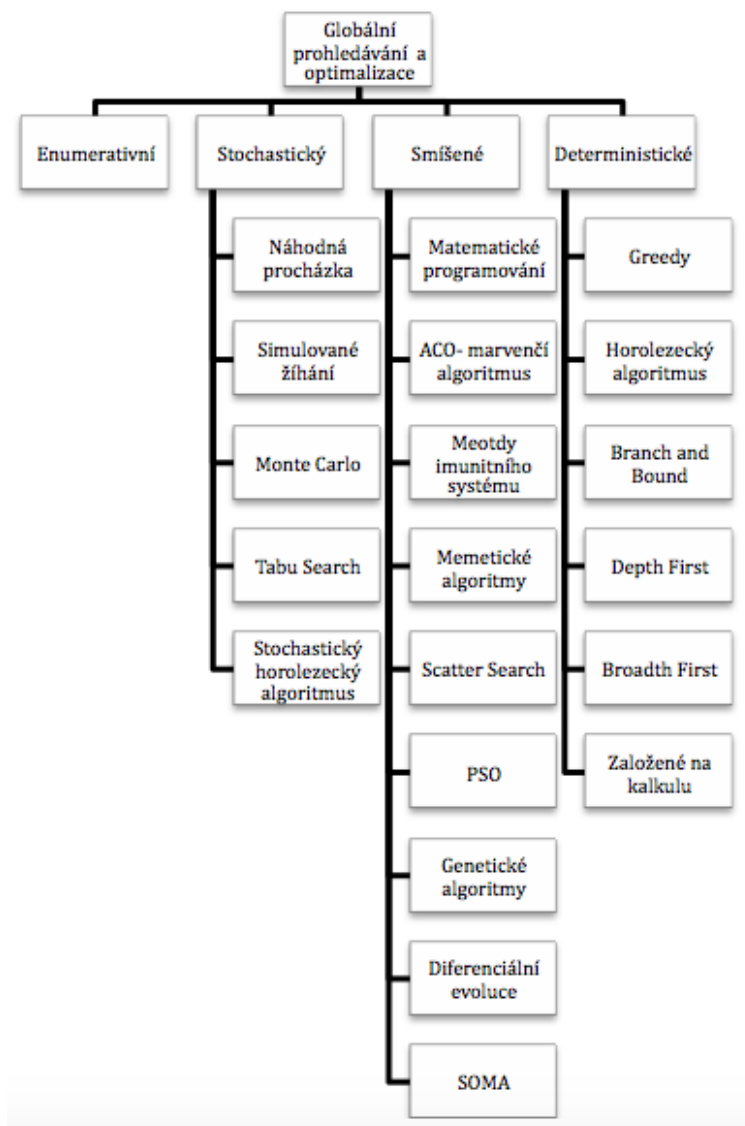
### 3. Metaheuristické

Deterministické

Pravděpodobnostní

Další možná rozdělení optimalizací jsou zobrazena na obrázcích [2.2](#) a [2.3](#).





Obrázek 2.3: Rozdělení podle (Zelinka et al., 2009)

## 2.3 Testovací funkce

Testovací funkce se používají na ověřování efektivity a zlepšování genetických algoritmů. Testování optimalizačních algoritmů může probíhat dvěma způsoby, za prvé se využívají úlohy, které již byly řešeny jiným algoritmem a tudíž se výsledky porovnají. Za druhé, se využívá množina umělých testovacích funkcí. Testovací funkce jsou zde rozděleny na neomezené multi-modální, neomezené uni-modální a funkce omezené hranicemi a z každé kategorie jsou uvedeny vybrané konkrétní funkce s grafickým znázorněním a definicí.

Testovací funkce jsou obvykle používány k vyhodnocení efektivity rozdílných prohledávacích algoritmů. Nicméně, výsledky vyhodnocení jsou závislé na testovacích problémech tak, jako jsou na algoritmech které jsou subjektem srovnání.

Testování je užitečný nástroj pro porozumění, jak a proč algoritmy fungují. Například testování může změřit jak výkon algoritmu závisí na účelových funkcích, ke kterým patří například dimenze, počet lokálních minim, stupeň závislosti parametru, dynamický rozsah parametrů, omezení, kvantování nebo zvuk. Testování také může ukázat, pro který kontrolní parametr kombinace je neefektivnější a to se stává velmi efektivním, když je vyhledávání efektivní sady kombinací kontrolních parametrů multi-objektivní optimalizační problém, ve kterém jsou rychlé a spolehlivé konvergence konfliktními cíli. Testováním algoritmů můžeme i porovnat jednotlivé výkony algoritmů mezi sebou. Navíc, testováním dochází k novým poznatkům, které následně slouží ke zlepšení optimalizačního výkonu. (Price et al., 2005)

### 2.3.1 Neomezené multi-modální testovací funkce

#### Ackleyho funkce

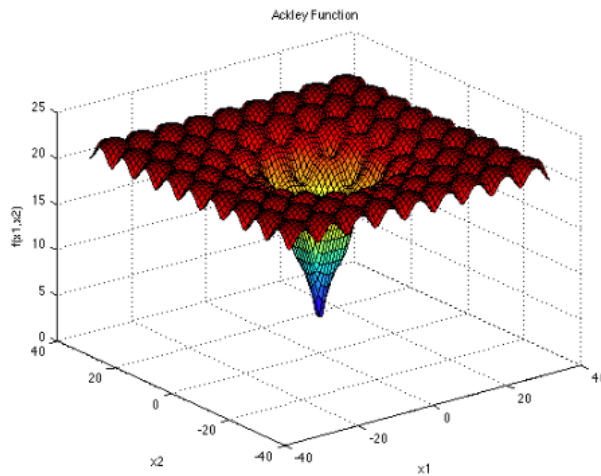
Ackleyova funkce je jedna z nejvíce citovaných multi-modálních testovacích funkcí. Zejména je využívána pro horolezecký algoritmus a algoritmy, kterým s vysokým rizikem uvíznutí v lokálním minimu. Funkce je zobrazena na obrázku 2.4.

$$f(x) = -a \exp \left( -b \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2} \right) - \exp \left( \frac{1}{d} \sum_{i=1}^d \cos(cx_i) \right) + a + \exp(1) \quad (2.3)$$

$$x_i \in [-32.768, 32.768] \quad (2.4)$$

#### Griewangkova funkce

Tato funkce je podobná Rastriginově funkci, z obrázku 2.5 je patrné, že má



Obrázek 2.4: Ackleyova funkce

mnoho rozsáhlých okálních minim, ačkoli jsou minima pravidelně rozděleny. Viz. rovnice 2.5 a 2.6.

$$f(x) = - \prod_{i=1}^{Dim} \cos \left( \frac{x_i}{\sqrt{i}} + \sum_{i=1}^{Dim} \frac{x_i^2}{4000} + 1 \right) \quad (2.5)$$

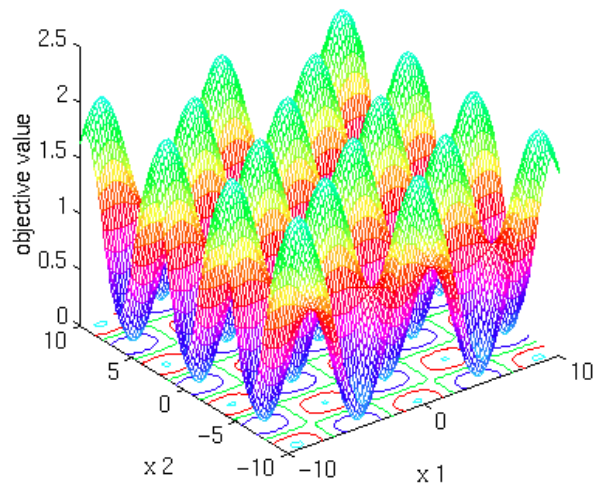
$$x_i \in [-600, 600] \quad (2.6)$$

### Rastriginova funkce

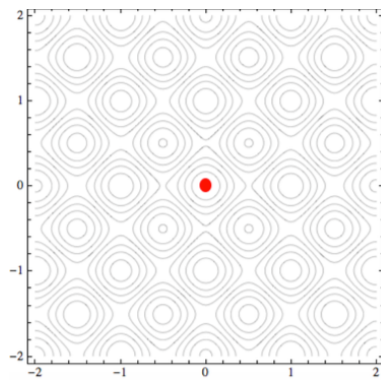
Rastriginova funkce má několik lokálních optim uspořádaných na straně velké deprese ve tvaru mísy. Tato funkce je oddělitelná a lehce řešitelná metodami, které mohou využívat rozložitelné funkce. To je mnohem těžší vyřešit, když se střídají. Rastriginova funkce je zobecněním dvou-dimenzionální funkce a symetrická okolo svých výsledků. Optimalizátory prohledávající sousedství průměrných vektorových populací, bude dobře fungovat, jelikož populace je zde dobře rozložena. Je to neomezená multi-modální testovací funkce. Viz. obrázky 2.6 a 2.7 (Price et al., 2005)

$$f(x_i |_{i=1, N}) = (N * 10) + \left[ \sum_{i=1}^N (x_i^2 - 10 \cos(2\pi x_i)) \right] \quad (2.7)$$

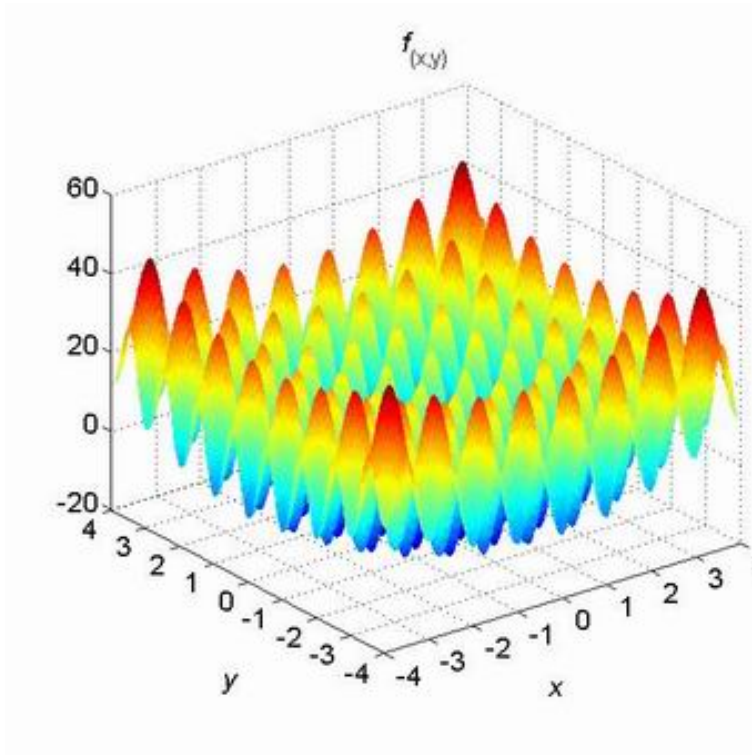
$$x_i \in [-5.12, 5.11] \quad (2.8)$$



Obrázek 2.5: Griewankova funkce



Obrázek 2.6: Rastriginova funkce. Bod uprostřed reprezentuje pozici globálního extrému



Obrázek 2.7: Rastriginova funkce

## 2.3.2 Neomezené uni-umodální testovací funkce

### Všeobecná Rosenbrockova funkce

Originální Rosenbrockova funkce byla pouze dvou-dimenzionální, ale později byla generalizována na multi-dimenzionální verzi. Hřbet na obrázu 2.8 ukazuje, že je tato uni-modální funkce nekonvexní. Tato funkce ukazuje limitovaný parametr závislý na pozici problému pro mnoho optimizérů. některé studie používají  $[-2.048, 2.048]$  a některé zase  $[-5.12, 5.12]$  pro výchozí hranice parametru. (Price et al., 2005)

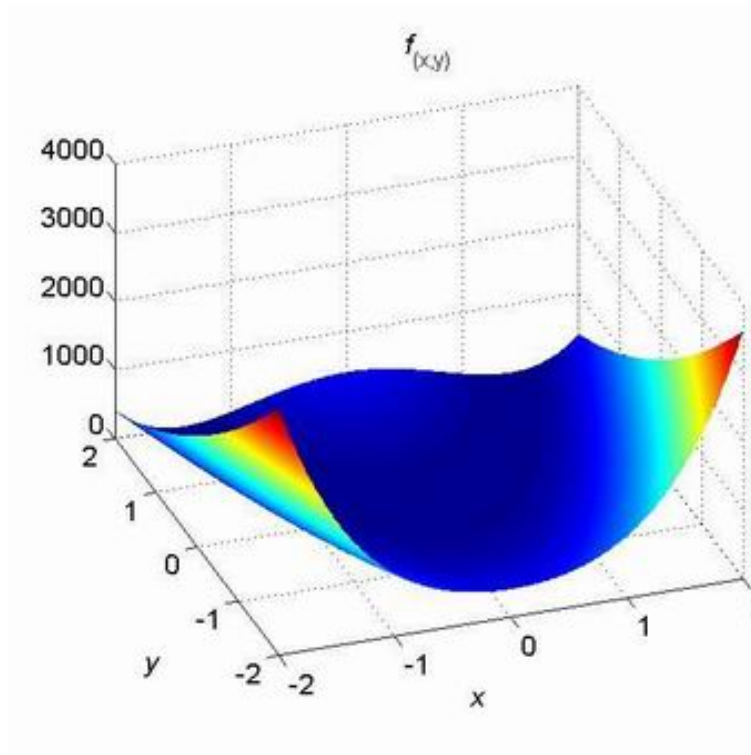
$$f_{x,y} = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \quad (2.9)$$

$$x_i \in [-5.12, 5.12] \quad (2.10)$$

$$x_i \in [-2.048, 2.048] \quad (2.11)$$

### De Jongovy funkce

Mezi nejpoužívanější testovací funkce lze zařadit De Jongovu první funkci



Obrázek 2.8: Rosenbrockova funkce

([De Jong, 1975](#)), která je standardem pro měření výkonnosti genetických algoritmů. Je to jednoduchá funkce testující lokální optimalizační rychlost prohledávací metody a mění rozměry. Je to funkce spojitá a konvexní.

De Jongův model testovací funkce byl modifikován a byl využit pro sestavení souboru benchmarkových funkcí. Tento testovací set obsahuje rozdílné charakteristiky, které mohou mít vliv na výkon algoritmu. Tento set nikdy nesloužil jako "zlatý standart", ale spíše byl navrhnout, aby ilustroval širokou účinnost genetických algoritmů pro rozdílné základní typy parametrů optimalizačních problémů. ([Belew, 1992](#)).

Následující rovnice zobrazují de Jongovy funkce, [2.12](#) unimodální funkce, [2.14](#) nelineární funkce, [2.16](#) nespojitá funkce, [2.18](#) noisy funkce, [2.20](#) multimodální funkce s několika lokálními optimy ([Whitley et al., 1996](#)).

$$f(x_i |_{i=1,3}) = \sum_{i=1}^3 x_i^2 \quad (2.12)$$

$$x_i \in [-5.12, 5.11] \quad (2.13)$$

$$f(x_i |_{i=1,2}) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2 \quad (2.14)$$

$$x_i \in [-2.048, 2.047] \quad (2.15)$$

$$f(x_i |_{i=1,5}) = \sum_{i=1}^5 [x_i] \quad (2.16)$$

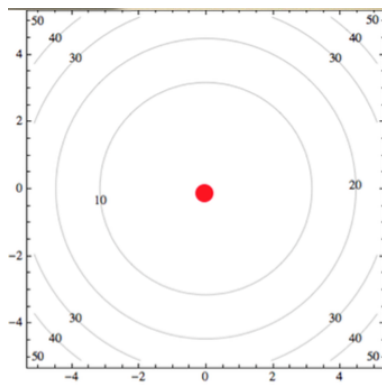
$$x_i \in [-5.12, 5.11] \quad (2.17)$$

$$f(x_i |_{i=1,30}) = \left[ \sum_{i=1}^{30} i x_i^4 \right] + \text{Gauss}(0, 1) \quad (2.18)$$

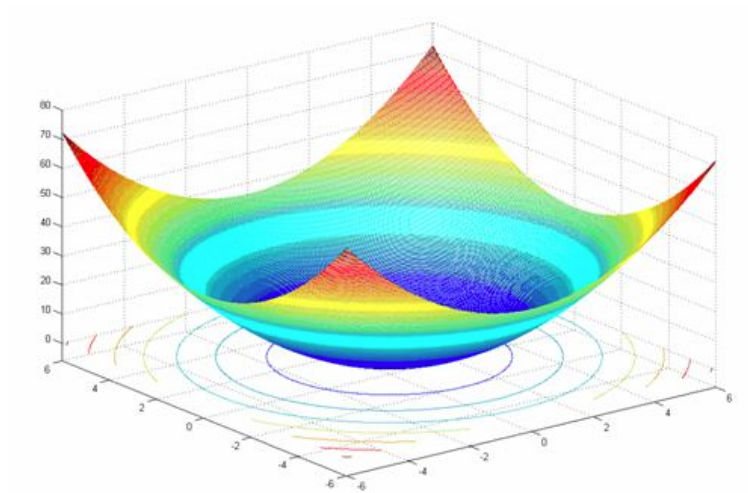
$$x_i \in [-1.28, 1.27] \quad (2.19)$$

$$f(x_i |_{1,2}) = \left[ 0.002 + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_i j)^6} \right]^{-1} \quad (2.20)$$

$$x_i \in [-65.536, 65.535] \quad (2.21)$$



Obrázek 2.9: První de Jongova funkce. Bod uprostřed reprezentuje pozici globálního extrému.



Obrázek 2.10: První de Jongova funkce

### 2.3.3 Testovací funkce omezené hranicemi

#### Schwefelova funkce

Tato funkce má pouze jeden lokální extrém, viz. obrázek 2.11.

$$\sum_{i=1}^{Dim} -x_i \sin\left(\sqrt[2]{|x_i|}\right) \quad (2.22)$$

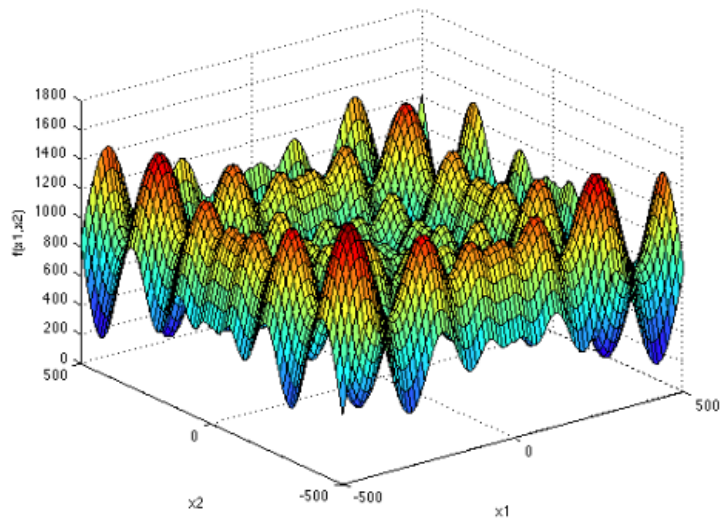
$$x_i \in [-512, 512] \quad (2.23)$$

#### Michalewiczova funkce

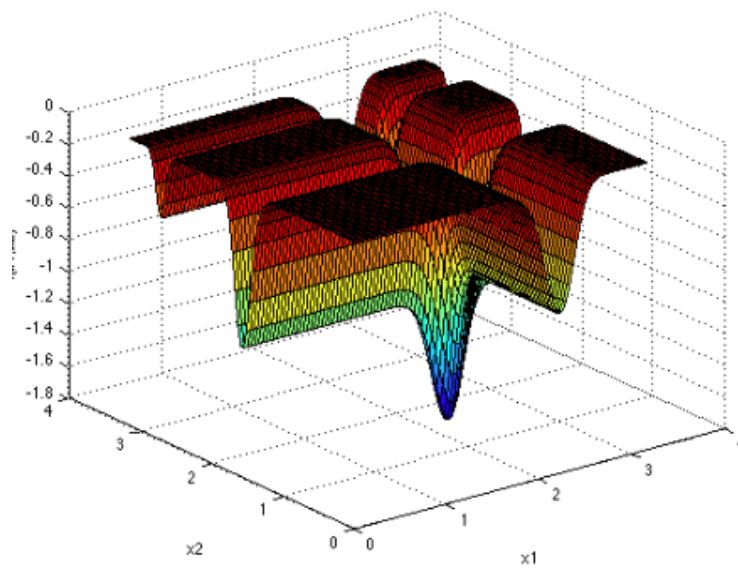
Tato funkce je pravděpodobně nejobtížnější z testovacích funkcí z pohledu optimalizace, má minimální informaci z gradientu funkce a to pouze v okolí lokálních a globálního extrému, viz. obrázek 2.12 (Pohl et al., 2014).

$$f(x) = -\sum_{i=1}^d \sin(x_i) \sin^{2m}\left(\frac{ix_i^2}{\pi}\right) \quad (2.24)$$





Obrázek 2.11: Schwefelova funkce



Obrázek 2.12: Michalewiczova funkce



## Kapitola 3

### PSO

PSO je stochastická optimalizační technika, která byla vyvinuta v roce 1995 Jamesem Kennedym a Russellem Eberhartem. Technika je založena na pozorování společenského chování zvířat jako např. hejna ptáků, roje včel nebo ryb. PSO vyhledává nejvhodnější oblasti v prohledávaném území za pomoci vzájemných vztahů mezi částicemi. (Kennedy – Eberhart, 1995)

PSO algoritmus simuluje chování ptačího hejna. Předpokládejme následující scénář: Skupina ptáků prohledává oblast a pátrá po nejvyšším vrcholku. Skupina neví, kde vrchol je, ale po každé iteraci ví, kdo našel zatím nejvyšší místo. Jak tedy najít vrchol? Nejlepší strategií by bylo následovat toho ptáka, který je v nejvyšší nadmořské výšce. (Zelinka et al., 2009)

System je inicializován populací náhodných řešení (jedinců) obdobně jako u genetických algoritmů a hledá optimální řešení vytvářením nových, lep-



Obrázek 3.1: Hejno. Ilustrační obrázek.

ších generací. PSO však nepracuje s evolučními operátory, jako je křížení či mutace narozdíl od genetických algoritmů. Potenciální řešení následují v řešeném prostoru trajektorie těch částic, které mají nejlepší vhodnost.

PSO vychází ze studia vzájemného chování a interakcí organismů, kteří vytvářejí roje, žijí v hejnech. Mnoho organismů tím, že vytvářejí hejna, nachází potřebné zdroje pro přežití. PSO vychází z Reynoldsova popisu vzájemných interakcí organismů v hejnu (intelligence hejna) (Reynolds, 1987). Jeho popis organizace skupiny organismů je spojen následujícími mechanismy:

- **separace** členová hejna zachovávají vzájemné postavení tak, aby při hledání optima v parametrickém prostoru nedocházelo k nežádoucímu překrývání
- **orientace** vzájemné postavení členů hejna je směřováno k nejlepšímu členu hejna
- **koheze** nové postavení při iteraci je ovlivněno blízkým okolím v hejnu (Máca, 2015).

### 3.1 Popis algoritmu

Algoritmus je přesný popis specifikující sled požadovaných kroků k vyřešení problému. V PSO každá částice prohledává multidimenzionální prostor řešeného problému. Částice má svoji polohu a rychlost a tyto informace si ponechává, aby se následující generace přiblížila hledanému optimu. Částice vzájemně spolupracují a na základě jejich spolupráce dochází k pokroku ve výpočtech.

Částice je náhodně umístěna v prostoru, má svoji rychlost a pozici, které jsou definovány jako vektor. Rychlost a pozice  $i$ -té částice představována jako

$$V_i = v_{i1}, v_{i2}, \dots, v_{iDim} \quad (3.1)$$

$$X_i = x_{1i}^t + V_i^{t+1} \quad (3.2)$$

Výkon každé částice je odhadnutý prostřednictvím specifického problému "goodness-of-fit" měřítka, který je základem pro zlepšení  $X_i$ . Nejlepší známá pozice  $i$ -té částice (personal/previous best) je představována jako  $P_i = p_{i1}, p_{i2}, \dots, p_{iDim}$  a nejlepší známá pozice uvnitř sousedství se nazývá *localbest* a je znázorňována jako

$$G = g_1, g_2, \dots, g_{Dim}. \quad (3.3)$$

Rychlost a pozice  $i$ -té částice je vylepšená dle následujících rovnic:

$$V_i^{t+1} = \omega V_i^t + c_1 U_1^t \otimes (P_i^t - X_i^t) + c_2 U_2^t \otimes (G^t - X_i^t) \quad (3.4)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (3.5)$$

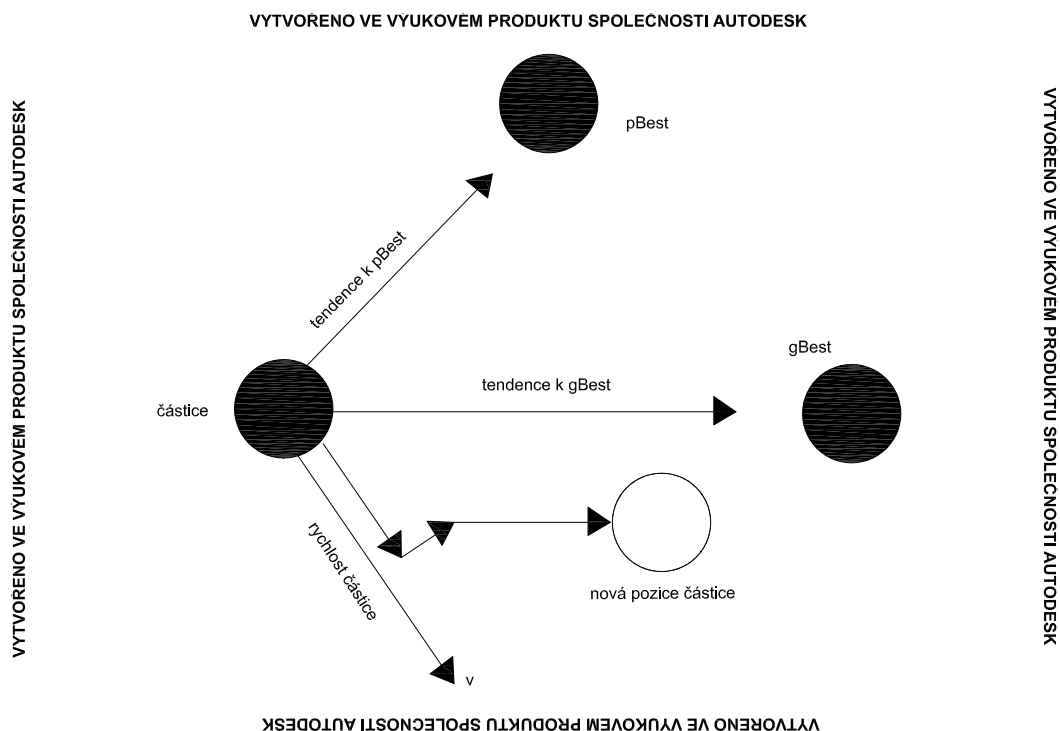
kde znamená  $i = 1, 2, 3, \dots, Dim$  a  $Dim$  je rovno velikosti hejna,  $t = 1, 2, 3, \dots, T$  a  $T$  je rovno maximálnímu počtu iterací.  $\omega$  je váha setrvačnosti, která však nebyla v původní verzi zahrnuta.  $c_1, c_2$  jsou koeficienty charakterizující kon-  
gativní a sociální chování.  $U_1$  a  $U_2$  jsou nezávislé a rovnoměrně rozšířené  
vektory v rozmezí  $[0,1]$ .

Z rovnic lze usoudit, že jedinci své hledání mohou ubírat třemi směry: individuální- pokračují svou cestou, konzervativní- vracejí se na svou dosud nejlepší pozici, přizpůsobivý- následují jedince, který zatím našel to nejlepší řešení. (Zelinka et al., 2009) Na obrázku 3.2 jsou znázorněny tyto tendence a skutečný pohyb částice.

Poté, co se částice přemístí na svou novou pozici, proces se opakuje znovu. Rychlost částice je však omezena maximální rychlostí  $V(max)$ , po přesáhnutí této rychlosti se vygeneruje nová rychlost nebo se vektor rychlosti ve směru, kde byla hodnota překročena, sníží na  $V_{max}$ . Předchází se tak tomu, aby se zabránilo částicím prudce zvyšovat svou rychlost a následnému přesáhnutí hranice prohledávané oblasti, když se tak stane částici se vygeneruje její nová pozice.

Pro správné fungování PSO je důležité vhodně nastavit tyto parametry: dimenze, rozsah, počet částic,  $V_{max}$ , učičí faktory  $c_1$ ,  $c_2$  a setrvačnost.

- **Dimenze** určuje problém, který optimalizujeme. Představuje počet argumentů účelové funkce, vektory pozice a rychlosti částice mají takový počet argumentů, jako je číslo dimenze.
- **Rozsah** velikost prohledávané oblasti
- **Počet částic** neboli velikost populace. Celkový počet jedinců, kteří prohledávají prostor, čím více částic obsahuje, o to bude prostor hustěji prohledán a tím i náročnější na výpočet a delší dobu pro výpočet. Počet částic by se měl pohybovat okolo  $10 \times Dim$ , obvykle 20 - 40 částic. Při větším počtu by nalezení trvalo podstatně delší dobu.
- $V_{max}$  maximální rychlost omezující rychlost částic



Obrázek 3.2: Pohyb částice ovlivněný jejími tendencemi (Zelinka et al., 2009), upraveno

- **Faktory**  $c_1$ ,  $c_2$  řídí kognitivní a sociální učení. Jsou to konstanty, které určují, nakolik je rychlost částice v každé iteraci ovlivněna hodnotami  $P_i$  a  $G$ .
- **Konstrikční faktor** ukazuje, jak správně nastavit proměnné  $\omega$ ,  $c_1$  a  $c_2$ , aby zaručily konvergenci a umožnily vynechat zpomalovací proměnnou  $V_{max}$ . (Eberhart – Shi, 2001) porovnávali chování hejna, když jedno využívalo hodnotu  $V_{max}$  a druhé faktor omezení. Faktor omezení sice konvergoval rychleji, ale občas uvízával v lokálním optimu. Nakonec se rozhodli pro využívání faktoru omezení i se zpomalením.
- **Váha setrvačnosti**  $\omega$  v původní verzi PSO nebyla setrvačnost ještě zavedena, až v roce 2001 v práci. Ve své práci se snaží nastavit tuto hodnotu, pohybující se v rozmezí  $[0,1.4]$  a nejlepší hodnoty se pohybují okolo hodnoty 1. Váha setrvačnosti byla navržena ke zlepšení kontrolovatelnosti dynamiky hejna a aby snížila důležitost těžko nastá-

vitelné maximální rychlosti. Když je  $\omega < 1$ , tak rychlost částice klesá a vyhledává lokální extrémy a naopak vyšší hodnota vyhledává globální extrémy. V poslední době dochází k novým návrhům, jak tuto proměnnou vylepšit, např. fuzzy systém nebo náhodné nastavení (Kennedy et al., 2001).

Špatné nastavení těchto parametrů může způsobit nestabilitu celého algoritmu. Příliš vysoké hodnoty těchto parametrů mohou způsobit, že částice nekontrolovatelně zrychlí namísto toho, aby důkladněji prohledávaly prostor v okolí svých hodnot  $P_i$  a  $G$ . Naopak příliš nízké hodnoty zvyšují náchylnost algoritmu k uvěznění v lokálním optimu (Pivoluska, 2008).

Zde je napsán algoritmus PSO v pseudokódu, je to variace, kdy částice nepřekročí maximální rychlost ani nepřekročí prohledávanou oblast:

- vstup
  - iterace: maximální počet iterací
  - částice: počet částic  $p_i$
  - gBest: nejlepší nalezené řešení v populaci
  - pBest: nejlepší nalezené řešení konkrétní částice
- **for** j=< počet částic **do**
  - begin**
  - $p_j = rand_j$
  - end**
- **for** i< iterace **do**
  - begin**
  - for** j=< počet částic **do**
    - begin**
    - (vypočítej rychlost částice podle)
    - (uprav pozici částice podle)
    - vhodnost =  $F_{cost}(p_i)$
    - if** vhodnost < pBest
    - then begin** pBest=vhodnost
    - end**
    - if** pBest < gBest
    - then begin** pBest=gBest
- **end**
- **end**

(Zelinka et al., 2009)



## 3.2 Topologie

Mezi dvě nejznámější topologie patří tzv. *gbest* a *lbest*. 3.3 V *gbest* populaci jsou trajektorie každé částice ovlivněny nejlepší polohou od každé částice z populace. Výhoda této topologie spočívá v rychlejší konvergenci a v jednom řešení z celé populace, tzv. nejlepší globální částice nebo-li *atraktant*, která k sobě přitahuje ostatní částice. Nevýhodu má však, když je tato jedna částice nesprávným řešením. U *lbest* topologie je trajektorie částice ovlivněna pouze svými dvěma sousedy, tzv. metoda lokálního prohledávání prostoru.  $K=2$ .

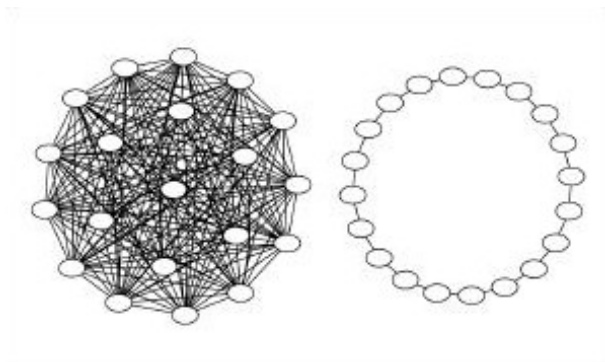
Obě topologie mohou být vnímány jako "sociální" susedství, protože vztahy mezi částicemi nezávisí na jejich pozici v prohledávaném prostoru, ale na "vnějších" vztazích, které nejsou závislé na problému, který se řeší. Susedství lze charakterizovat z hlediska dvou faktorů:

- $K$  stupeň propojitelnosti
- $C$  množství shlukování, měří počet sousedů částice (Kennedy – Mendes, 2002).

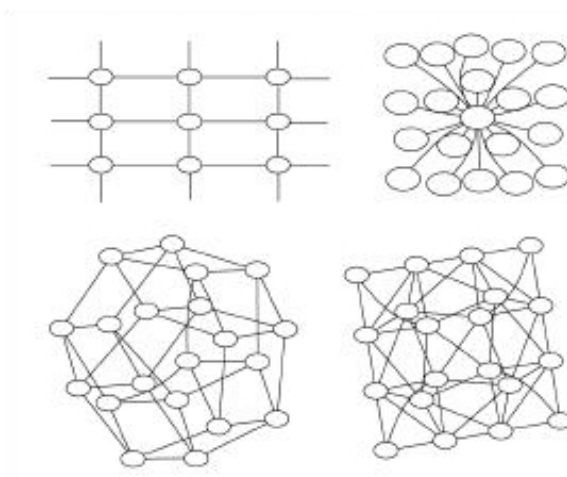
Mezi další používané topologie patří: von Neumann, náhodná, pyramida, hvězda a heterogenní topologie. 3.4

- von Neumann publikovali Kennedy a Mendes. Částice jsou propojeny přes dvourozměrnou mřížku, každá částice má tedy 4 sousedy,  $K=4$  (nahore, dole, vlevo, vpravo). Možnost konvergence je zde pomalejší.
- pyramida trojdimenzionální trojúhelníková mřížka  $K=3$
- hvězda Hvězdíková topologie připomíná *gbest* topologii, vybere se jedna částice, která je ve středu a na ni jsou poté napojeny všechny ostatní částice. Může zde docházet k předčasně konvergenci.
- heterogenní Částice jsou rozděleny do několika seskupení.

Mimo všech kladů, které PSO má, má i své slabé stránky. Obvykle se stává, že rychlost částic je příliš rychle vygenerována a částice se potom příliš vzdalují od nejlepšího řešení. Při složitějších problémech s několika optimy dochází k předčasné konvergenci.



Obrázek 3.3: Grafické znázornění *gbest* a *lbest* topologie



Obrázek 3.4: Grafické znázornění topologií zleva: von Neumann, hvězda, trojrozměrná von Neumann, pyramida

# Kapitola 4

---

## MODIFIKACE PSO

Od doby, kdy byl algoritmus PSO v roce 1995 vydán byla vyvinuta snaha o zlepšení jeho optimalizační schopností, např. už zmiňovaná váha setrvačnosti, která se objevila o šest let později. V následující kapitole bych ráda uvedla pár příkladů některé z modifikací algoritmu PSO.

### 4.1 Celulární optimalizace hejnem částic

Sloučení celulární automaty (dále jen CA) s optimalizací hejnem částic bylo publikováno v článku ([Shi et al., 2011](#)), je zde navrženo, aby se mechanismus CA integroval do rychlosti a modifikoval trajektorie částic, aby se částice vyhnuly uvíznutí v lokálním minimu. Byly zde představeny dvě verze, CPSO- vnitřní a CPSO- vnější.

CPSO- vnitřní používá informace uvnitř hejna částic ke spolupráci tak, tím že zvažuje každou částici jako buňku, kdežto CPSO- vnější umožňuje buňkám, které patří do hejna komunikovat s buňkami mimo hejno. Každé potenciální řešení je označeno jako inteligentní buňka.

Ačkoli CA a PSO jsou rozdílné, přesto se v několika prvcích shodují. Oba dva algoritmy jsou složeny z řady jedinců-buňky a částice a také u obou probíhá vzájemná komunikace k předání potřebné informace. U CA, každá buňka komunikuje se svými sousedy a změní svůj stav buňek s přihlédnutím k současnemu stavu buňek, takováto komunikace probíhá i mezi částicemi v hejnu.

Byly navrženy tři typické mřížkové struktury CA, které byly použity jako sousedství, umožňující částicím komunikovat uvnitř hejna. Byla také navržena nová strategie CA, která je založená na principu "inteligentní- buňka" a částice zaměstnávají informace z vnějšku hejna. Teoretické studie byly udělány k analýze konvergence CPSO a výpočetní experimenty jsou prováděny k porovnání navrženého algoritmu s jinými variantami PSO. Podle výsledků experimentu byla navržena metoda lepší než ostatní varianty PSO.

## 4.2 Centrální optimalizace hejnem částic

V další studii je představen algoritmus centrální optimalizace hejnem částic (CenterPSO), kde středová částice je zahrnuta do lineárně klesající váhy PSO (LDWSPO- z angl. Linearly weight particle swarm optimization). Na rozdíl od běžných částic v LDWSPO, středová částice nemá žádnou explicitní rychlost a je nastavena na střed hejna v každé iteraci. Ostatní aspekty středové částice jsou stejné jako u běžné částice. Jelikož střed hejna je příznivá pozice, středová částice má dobrou fitness hodnotu. Ještě důležitější je, že vzhledem se často vyskytuje jako nejlepší částice hejna, přitahuje další částice a vede vyhledávání směru celého hejna. Navíc jsou Center PSO a LDWSPO jsou porovnávány jako trénující algoritmy neuronové sítě. Center-PSO je ve výsledcích výkonnější než LDWSPO (Liu et al., 2007).

## 4.3 Adaptivní optimalizace hejnem částic

(Zhan et al., 2009) ve své studii představují lepší vyhledávací účinnost než je v klasické PSO. Adaptivní PSO (dále jen APSO, z anglického- Adaptive Particle Swarm Optimization) se skládá ze dvou důležitých kroků. Možná ještě důležitější než účinnost vyhledávání je, že je schopná provést prohledávání v celém prostoru s vyšší rychlostí konvergence. První krok APSO je, že vyhodnotí rozložení populace částic a fitness hodnoty. Výsledky dokazují, že podstatně zvyšuje výkon paradigmatu PSO, hlavně co se týká rychlosti konvergence, globální optimality a přesnosti a spolehlivosti algoritmu. APSO představuje dva nové parametry PSO.

## 4.4 Optimalizace Multi-hejnem

Optimalizace multi-hejnem optimalizace je další varianta PSO, která využívá místo jednoho hejna více sub-hejn, každé toto sub-hejno se zaměřuje na jiný prohledávací prostor. Multi-swarm PSO je speciálně uzpůsobené pro optimalizaci na multi-modální problémy, kde existuje více lokálních optim.

Při řešení multi-modálních problémů je důležité dosáhnout rovnováhy mezi explorací a exploatací a právě tuto rovnováhu se snaží poskytnout multi-swarm. Explorace znamená prohledávání řešeného prostoru s cílem nalézt vhodné optimum. Exploatace je důkladné prohledávání předběžně nalezených optim s cílem je přesně lokalizovat (Němečková, 2013). Snaží se vyvážit vztah mezi explorací a exploatací, aby nedocházelo k oslabení pro-

cesu vyhledávání. Postup multi-hejna je rozdělen do několika fází. Každá fáze je více zaměřena buď na exploataci nebo exploraci.

Koordinace sub-hejn závisí na specifické diverzifikační metodě, kterou zavádí metoda multi-hejna. Hlavním bodem této optimalizace je, že počáteční rychlost a počáteční pozice nejsou vybrány náhodně, ale ponechávají si informace z předchozích trajektorií částic.

## 4.5 Další vybrané modifikace PSO

- **Druhov**á PSO (*Speciation* PSO) V populaci jsou částice odlišné, každá částice jiný druh (*species*). Odlišují na základě morfologie nebo biologie. Po první iteraci jsou vybrány nejlepší částice ze svých skupin a stávají se vůdci, jsou nazývány "semínka" (*seeds*). Po každé iteraci se vůdci mohou střídat. **INPSO** (Independent Neighbourhoods PSO) rozděluje hejno do několika menších hejn, která jsou na sobě nezávislá.
- **Hybridní PSO** kombinuje INPSO s evolučním algoritmem GPEA (Geometrical Place Evolutionary Algorithm). Řešení je nalezeno za delší čas.

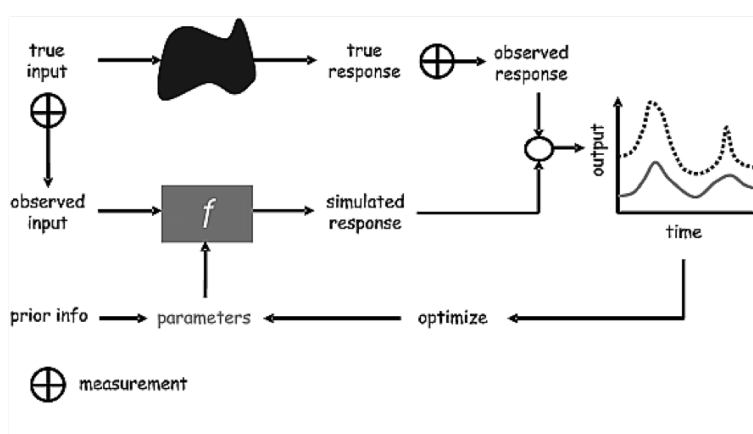


# Kapitola 5

## PSO V HYDROLOGII

Matematické modely jsou užitečné v různých vědních a inženýrských oborech, jako je biologie, fyzika, chemie, enviromentální vědy, medicína a ekonomie (Wu et al., 2014). V hydrologii slouží PSO a jiné optimalizační metody k odhadu parametrů hydrologických modelů, které se používají k predikci odtoku, posouzení srážkovo-odtokových procesů, meteorologické předpovědi. Zvolený model by měl co nejvěrněji napodobovat skutečnost, k tomu je důležité správné nastavení parametrů a určit optimalizační kritérium.

(Vrugt et al., 2008) schematicky vymezují rámec studia hydrologického systému atributy zobrazenými na obrázku 5.1. Popis vzájemného vztahu mezi hydrologickým systémem povodí, jeho reálným vstupem a skutečným výstupem vody je prováděn prostřednictvím popisu vztahu mezi měřeným vstupem vody, modelem hydrologického systému, modelovým a měřeným výstupem vody. Hlavním cílem je kalibrace hydrologického modelu, jejíž výsledkem je co nejbližší možné přiblížení modelované a pozorované časové řady fyzikální veličiny, které charakterizují odtok z povodí.(Máca, 2015)



Obrázek 5.1: Kalibrace hydrologického modelu<sup>a</sup>

<sup>a</sup>Převzato z (Vrugt et al., 2008)

## 5.1 Aplikace na povodí

(Chau, 2006) se zabývá přesnou předpovědí vodní hladiny, která je velmi důležitou při vydávání varování před hrozící povodní a následnou evakuací v případě potřeby. Stávající metody včetně modelování srážko-odtokového procesu nebo statistických technik nesou exogenní vstupní data společně s počtem předpokladů. Využití modelu umělých neuronových sítí se ukázalo jako hospodárná technika, ale jejich trénování, obvykle na algoritmu se zpětnou propagací chyby, má pomalou konvergenci a Zde je představen model PSO, který je připraven trénovat perceptrony. Přístup je aplikován k předpovědi vodní hladiny na řece Shing Mun v Hong Kongu s různými časy na základě měření na horním proudu řeky nebo jsou použita data z různých stanic.

Dle výsledků studie je pravděpodobné, že je možné dosáhnout větší přesnosti za kratší čas narozdíl od metody zpětného výběru (back-propagation). Z trénování a ověřování simulace je zřejmé, že výsledky předpovědi vodní hladiny jsou přesnější než běžně používaný vycházející perceptron (based perceptron). Výchozí výsledek ukazuje, že technika PSO může vystupovat jako alternativa trénujícího algoritmu pro model umělé neuronové sítě. Toto je autorova pilotní studie, důkladnější studie budou zpracovány v budoucích pracích.

(Liang – Suganthan, 2005) použili PSO pro kalibraci parametrů modelu HIMS (hydroinformatic modeling system). Ve své studii srovnávají základní algoritmus PSO s metodami využívající komplexity a míchání. Model HIMS má dohromady devět parametrů např.: infiltrační koeficient, evapotranspirační koeficient, parametr obohacování podzemních vod, které se musí optimalizovat. Studie byla prováděna na povodí řeky Liu v Číně. Výsledky této práce dokazují, že když se PSO rozdělí do několika menších sub-hejn, výrazně se zvýší výkonnost modelu a je spolehlivým nástrojem pro kalibraci parametrů hydrologického modelu. (Němečková, 2013)

Další studii provedli Mauricio Zambrano-Bigiarini a Rodrigo Rojas v roce 2013 (Zambrano-Bigiarini – Rojas, 2013), kde navrhli balíček hydroPSO. Více o této studii je uvedeno v následující kapitole.

## 5.2 HydroPSO

HydroPSO je software používaný ve výpočetním programu R, který vynalezl Mauricio Zambrano-Bigiarini a Rodrigo Rojas v roce 2012. Je to software, který je hlavně zaměřen na analýzu citlivosti, kalibraci parametrů hydrologického modelu nebo vyhodnocení výsledků kalibrace pro životní prostředí.



HydroPSO zavádí několik nejmodernějších zlepšení a zdokonaluje nastavení PSO algoritmu. (Zambrano-Bigiarini – Rojas, 2013)

R je software, který se používá pro analýzu dat. Je využíván hlavně pro statistiku, vizualizaci dat (grafy) a pro matematické modelování. Obsahuje všechny standardní statistické metody a většina nově vzniklých statistických a modelovacích technik vzniká právě v R. Tento software je nabázi open-source, tudíž je možné jej stáhnout a užívat zdarma. Byl vytvořen v roce 1993 Ross Ihakkou a Robertem Gentlmanem.

### 5.2.1 Popis balíku hydroPSO

HydroPSO je multi-OS a na modelu nezávislý balík, který byl navrhnout, aby umožnil uživateli vytvořit analýzu citlivosti (udává, jak se změní výsledek, když dojde ke změně odhadu vstupních předpokladů, modelování a stanovení rizika), kalibraci modelu a vyhodnocení výsledků. HydroPSO není omezen počtem hydrologických modelů, může být propojen s různými modely s poměrně nízkou námahou, je plně komaptibilní s kalibračními nástroji uplatňující PEST jako předlohu a umožňuje podobnost.

Jediné, co hydroPSO potřebuje znát jsou parametry modelu, který má být kalibrován, kam mají být zapsány a odkud. Kalibrace hydroPSO komunikuje s několika modely přes ASCII složky nebo R obalové (wrapper) funkce, které načtou vstupy a výstupy a vypočítávají výkon modelu, pokudalgoritmu, je vyžadován.

Ve studii bylo aplikováno deset testovacích funkcí na šesti optimalizačních algoritmech, byly zvoleny tak, aby výsledky nebyly zkresleny směrem k dané množině povrchů odezvy a zhodnotily výkon algoritmů, když není optimum lokalizováno v původních souřadnicích. Zde jsou vypsány algoritmy:

- Gauss- Marquardt-Levenberg (GML) (Marquardt, 1963),
- Shuffled Complex Evolution (SCEUA) (Duan et al., 1992),
- Diferenciální evoluce (DE) (Storn – Price, 1997)
- Differential Evolution Adaptive Metropolis (DREAM) (Vrugt et al., 2011)
- SPSO-2011 (Clerc, 2012)
- hydroPSO.

Funkce byly anlyzovány v 10-, 20- a 30- rozměrném prostoru. Zde je přehled objektivních funkcí:

- Sphere function
- Rosenbrockova funkce
- Rastriginova funkce
- Griewangkova funkce
- Ackley function
- Schafferova F6 funkce
- Shifted Sphere function
- Shifted Griewank function
- Shifted Ackley function
- Shifted Rastrigin function

Z výsledků srovnání je patrné, že žádný algoritmus není ten nejlepší, avšak algoritmus hydroPSO se projevilo být více účinnější než ostatní algoritmy, ukázalo se že je efektivnější než ostatní algoritmy v 9 z 10 funkcí a také se nejvíce přibližuje k hledané optimální hodnotě.

### 5.2.1.1 Funkce hydroPSO

Mezi vlastnosti funkcionality hydroPSO patří:

- **lhoat** zařazuje funkci Latin Hypercube One-factor-At-a-Time (LHOAT) pro citlivostní analýzu. Lhoat() vytváří pořadí parametrů. Parametr s největším vlivem získá označení 1 a ostatní, s menším efektem, získají označení podle celkového počtu parametrů.
- **hydromod** je jedna z klíčových složek hydroPSO. Spouští uživatelem nedefinovaný model, který má být kalibrován- optimalizován a vrátí hodnotu goodness-of-fit jako měřítko výkonu modelu porovnáním pozorovaných dat se simulovanými
- **hydroPSO** kalibrace kódů modelu v reálném prostředí state-of-the-art verze PSO,
- **plotresults** vykresluje grafy na základě výsledků z hydroPSO

- **verification** spouští model a získává hodnotu goodness-of-fit porovnáním simulované hodnoty a pozorování pro optimální nastavení parametrů nalzených optimalizací
- **PEST2hydroPSO** import výchozích složek PEST (.pst, tpl., ins.) do hydroPSO
- **hydroPSO2PEST** export obashu výchozích složek hydroPSO do PEST
- **testfunctions** testovací funkce běžně používané jako benchmark pro globální optimalizace
- **readresults**

#### 5.2.1.2 Případové studie

HydroPSO byl aplikován ke kalibraci hydrologického modelu SWAT-2005. SWAT (akronym pro Soil and Water Assessment Tool) je model, který byl vyvinut pro předpověď vlivu půdního uspořádání na vodu, sediment a zemědělské pesticidy ve velkém měřítku rozvodí s různými půdami, land-use s ohledem na dlouhou časovou periodu (Neitsch et al., 2005). Dále byl aplikován na model MODFLOW-2005.

Algoritmus hydroPSO byl aplikován na dvě území. V prvním případě se jedná o povodí řeky Egy ve Španělsku a ve druhé studii je aplikován na zvoděň Pampa del Tamarugal v Chile.

Algoritmus hydroPSO byl použit pro kalibraci hydrologického modelu SWAT na povodí řeky Egy, která je přítokem řeky Ebro v provincii Navarra, Španělsko. Rozloha je  $1445 \text{ km}^2$  a nadmořská výška v rozmezí 300 až 1400 m.n.m. Na zrealizování hydroPSO bylo vybráno vyšší povodí Egy, které má rozlohu  $808 \text{ km}^2$ , průměrné roční srážky jsou 818 mm a průměrný denní průtok měřený v Estelle je  $12,5 \text{ m}^3/\text{s}$  měřený v časovém rozmezí 1961-1990.

Druhá studie, kde algoritmus hydroPSO kalibruje hydrologický model proudění podzemní vody MODFLOW pro zvoděň Pampa del Tamaragal v Chile. Zvoděň leží v poušti Atacama a rozkládá se na  $5000 \text{ km}^2$  hydroPSO zde bylo pužito pro kalibraci modelu MODFLOW- 2005, 30 parametrů bylo propojeno, aby obnovily evapontranspiraci, transpiraci, stav hranic a hydraulickou vodivost.

Výsledky obou studií dokazují, že hydroPSO je efektivní model s velmi dobrými optimalizačními schopnostmi.

HydroPSO ukazuje několik inovativních aspektů, např. možnost vykonávání citlivostní analýzy používající stejný balík a automatizaci vykreslování grafů pro analýzu výsledků. Výsledky studie zdůrazňují význam správného výběru goodness-of-fit měřítka. Případové studie ilustrují vynikající stabilitu zvládnout několik problémů najednou.

Verze hydroPSO, která je zde popsána, funguje zatím pouze jako jednoúčelová, víceúčelová optimalizace a analýzy globální citlivosti jsou předmětem dalšího vývoje. Aplikace se týkají pouze hydrologických modelů, avšak na základě pružnosti tohoto softwarového nástroje je možné jej aplikovat na širší třídu modelů. ([Zambrano-Bigiarini – Rojas, 2013](#))

## Kapitola 6

---

### DISKUSE

Tato literární rešerše shrnuje obecné poznámky ohledně optimalizačních algoritmů a popis optimalizačního algoritmu PSO, jeho modifikaci a implementaci v hydrologii. Rozdělení optimalizačních algoritmů není jednoznačné, jelikož několik autorů má pokaždé jiné rozdělení, v této práci jsem se snažila o srozumitelné rozdělení a je zde uvedeno několik typů na rozdělení dle různých autorů.

Popsání základního algoritmu PSO nebylo složité, avšak různé modifikace tohoto algoritmu byly celkem rozsáhlé. Modifikací tohoto algoritmu se objevuje v posledních letech stále více a ve studiích autoři i nadále pokračují ve vylepšování, ikdyž ve většine studiích dosahoval tento algoritmus dobrých výsledků, je to perspektivní optimalizační metoda využívaná nejen v hydrologii.



# Kapitola 7

## ZÁVĚR

Cílem této práce bylo rešeršně zpracovat optimalizační metodu PSO. K porozumění, jak fungují optimalizační metody, hlavně PSO, byl popsán základní algoritmus PSO, princip na kterém pracuje a dále, jak byl vylepšován a modifikován, aby se předcházelo nechtěným chybám, aby bylo jasné k čemu je možné PSO využít v hydrologii, jsou v poslední části uvedeny některé konkrétní implementace v hydrologii.

V první části je prezentováno rozdělení optimalizací a testovacích funkcí, které slouží k ověřování výkonnosti, výhod a hlavně pomáhají ve vylepšování algoritmů. Jelikož PSO byla popsána již v roce 1995, bylo od té doby zpracováno mnoho modifikací ve snaze vylepšit výkonnost, vybrané modifikace jsou uvedeny ve čtvrté kapitole. Celulární PSO, kde jde o spojení PSO a celulární automaty, se snaží vylepšit zabránění uvíznutí v lokálním minimu a v testování se tato modifikace ověřila. Jedna z dalších modifikací je centrální PSO, která byla srovnána s LDWSPO a prokázala se být výkonější.

V závěru byla popsána implementace PSO v hydrologii, je zde popsán software hydroPSO, který je určen na citlivostní analýzu, kalibraci modelu a vyhodnocení výsledků. HydroPSO byl použit pro kalibraci hydrologického modelu SWAT a v druhé studii byl využit na kalibraci hydrologického modelu proudění podzemní vody MODFLOW. Výsledky obou studií hodnotí hydroPSO jako efektivní model s velmi dobrými optimalizačními schopnostmi a ve srovnání s ostatními algoritmy byl nejlepší.

Během zpracování bakalářské práce jsem získala přehled o optimalizacích, jejich širokém využití a neustálém vylepšování PSO. Další studie o PSO budou pravděpodobně rozšířeny v diplomové práci.





# LITERATURA

- Belew, R. Paradigmatic over-fitting. *GA- Digest 6 (18)*. 1992. [2.3.2](#)
- Chau, K. Particle swarm optimization training algorithm for {ANNs} in stage prediction of Shing Mun River. *Journal of Hydrology*. 2006, 329, 3–4, s. 363 – 367. [1](#), [5.1](#)
- Clerc, M. Standard Particle Swarm Optimisation. 15 pages, September 2012. [5.2.1](#)
- De Jong, K. A. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD thesis, University of Michigan Ann Arbor, MI, USA, Ann Arbor, MI, USA, 1975. AAI7609381. [2.3.2](#)
- Duan, Q. – Sorooshian, S. – Gupta, V. Effective and efficient global optimization for conceptual rainfall-runoff models. *Water Resources Research*. 1992, 28, 4, s. 1015–1031. [5.2.1](#)
- Eberhart, R. – Shi, Y. Particle swarm optimization: developments, applications and resources. *Evolutionary Computation*. 2001, 1, s. 81–86 vol. 1. [3.1](#)
- Kennedy, J. – Eberhart, R. Particle swarm optimization. In *IEEE International Conference on Neural Networks Proceedings*, s. 1942–1948. IEEE, Neural Networks Council, I E E E, 1995. [3](#)
- Kennedy, J. – Mendes, R. *Population structure and particle swarm performance*. 2. IEEE computer Society, 2002. [3.2](#)
- Kennedy, J. – Eberhart, R. C. – Shi, Y. Swarm intelligence. *Kaufmann, San Francisco*. 2001, 1, s. 700–720. [3.1](#)
- Liang, J. – Suganthan, P. Dynamic multi-swarm particle swarm optimizer. In *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE*, s. 124–129, 2005. [5.1](#)
- Liu, Y. et al. Center particle swarm optimization. *Neurocomputing*. 2007, 70, 4–6, s. 672 – 679. [4.2](#)

- Máca, P. Hydrologické modely a automatická optimalizace parametrů. *ČZU Praha*. 2015. [3](#), [5](#)
- Mañas, M. *Optimalizační metody*. SNTL- Nakladatelství technické literatur, 1979. [1](#), [2.1](#)
- Marquardt, D. W. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics*. 1963, 11, 2, s. 431–441. [5.2.1](#)
- Mendes, R. *Population topologies and their influence in particle swarm performance*. PhD thesis, Citeseer, 2004. [2.2](#)
- Neitsch, S. et al. Soil and Water Assessment Tool. *Blackland Research Center Texas Agricultural Experiment Station*. 2005. [5.2.1.2](#)
- Němečková, M. *Aplikace optimalizačních metod v hydrologickém modelování*. PhD thesis, Česká zemědělská univerzita v Praze, 2013. [4.4](#), [5.1](#)
- Pivoluska, M. *Algoritmus roja částic PSO a demonstrácia jeho vlastností*. PhD thesis, Masarykova Univerzita Fakulta Informatiky, 2008. [3.1](#)
- Pohl, V. – Jirsík, V. – Honzík, P. Optimalizační algoritmy s pravděpodobnostním směrovým vektorem a jejich srovnání s principiálně podobnými metodami. *Elektrorevue*. 2014, 16, 1. [2.3.3](#)
- Price, K. V. – Storn, R. M. – Lampinen, J. A. *Differential Evolution A Practical Approach to Global Optimization*. Springer, 2005. [2.3](#), [2.3.1](#), [2.3.2](#)
- Reynolds, C. W. Flocks, Herds and Schools: A Distributed Behavioral Model. *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*. 1987, s. 25–34. [3](#)
- Riget, J. – Vesterstrom, J. S. A Diversity- Guided Particle Swarm Optimization- the ARPSO. *EVALife Project Group*. 2002. [2.2](#)
- Shi, Y. et al. Cellular particle swarm optimization. *Information Sciences*. 2011, 181, 20, s. 4460 – 4493. [4.1](#)
- Storn, R. – Price, K. Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*. 1997. [5.2.1](#)

- Vrugt, J. – Braak, C. – Diks, C. Accelerating Markov Chain Monte Carlo Simulation by Differential Evolution with Self-Adaptive Randomized Subspace Sampling. *International Journal of Nonlinear Sciences and Numerical Simulation*. 2011, 10, s. 273–290. [5.2.1](#)
- Vrugt, J. A. et al. Treatment of input uncertainty in hydrologic modeling: Doing hydrology backward with Markov chain Monte Carlo simulation. *Water Resources Research*. 2008, 44, 12, s. n/a–n/a. [5](#), [a](#)
- Weise, T. Global Optimization Algorithms- Theory and Application. *psu.edu*. 2008. [2.2](#), [2.1](#)
- Werner, T. Optimalizace Zárodek skript k předmětu A4B33OPT, 2011. [2.1](#)
- Whitley, D. et al. Evaluating Evolutionary Algorithms. *Artificial Intelligence*. 1996, 85, 1-2, s. 245–276. [2.3.2](#)
- Wu, Y. – Liu, S. – Yan, W. A universal Model-R Coupler to facilitate the use of R functions for model calibration and analysis. *Environmental Modelling Software*. 2014, 62, 0, s. 65 – 69. [5](#)
- Zambrano-Bigiarini, M. – Rojas, R. A model-independent Particle Swarm Optimisation software for model calibration. *Environmental Modelling Software*. 2013, 43, 0, s. 5 – 25. ISSN 1364-8152. doi: <http://dx.doi.org/10.1016/j.envsoft.2013.01.004>. [1](#), [5.1](#), [5.2](#), [5.2.1.2](#)
- Zelinka, I. et al. *Evoluční výpočetní techniky- principy a aplikace*. BEN-technická literatura, 2009. ISBN 978-80-7300-218-3. [1](#), [2.2](#), [2.3](#), [3](#), [3.1](#), [3.2](#), [3.1](#)
- Zhan, Z.-H. et al. Adaptive Particle Swarm Optimization. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*. Dec 2009, 39, 6, s. 1362–1381. ISSN 1083-4419. doi: [10.1109/TSMCB.2009.2015956](https://doi.org/10.1109/TSMCB.2009.2015956). [4.3](#)