National Conference with International Participation
ENGINEERING MECHANICS 2007
Svratka, Czech Republic, May 14 – 17, 2007

SYSTEM AND PROCES OPTIMIZATION BY SWARM BASED OPTIMIZATION

L. Fišerová, L. Raudenská*

Summary: *Swarm intelligence is based on nature-inspired behavior and is successfully applied to optimization problems in a variety of fields. The goal of optimization is to find the optimum in the smallest possible amount of iterations where the optimum means the best among all possibilities chosen from a particular point of view (so-called: criterion).*

Keywords: *System diagnostic, Swarm-based optimization, Bees algorithm.*

Úvod

Optimalizační problémy se vyskytují téměř v každém oboru lidské činnosti. Každodenně je řešena řada problémů, jak něco provést nejlepším způsobem. Ovšem ne u všech metod je možno využít tradičních metod optimalizace, které jsou určeny především pro exaktní algoritmy. Když ovšem tento algoritmus není znám, či jeho řešení by bylo příliš rozsáhlé, využívají se v praxi postupy, které dokáží v krátkém výpočetním čase najít sice ne nejdokonalejší, ale velmi kvalitní řešení.

V následujících odstavcích jsou rozebrány optimalizační metody Rojové inteligence, které jsou inspirovány chováním skutečných biologických systémů.

1 Swarm-based optimisation - Rojová inteligence

Rojová inteligence je technika umělé inteligence založená na studiu kolektivního chování samoorganizujících se systémů.

Systémy rojové inteligence se obvykle skládají z populace jedinců, kteří na sebe vzájemně působí a vznikají také interakce s okolním prostředím. Jednotlivci spolu mohou komunikovat přímo, nebo nepřímo působením v místním prostředí. Ačkoli tyto systémy nemají žádnou centrální kontrolu chování jednotlivců, vzájemné působení mezi jednotlivci a jednoduché vzory chování jednotlivců obvykle vedou k objevení úhrnného chování typického pro celou kolonii. Toto může být v přírodě pozorováno například u mravenců, včel, ptáků nebo bakterií.

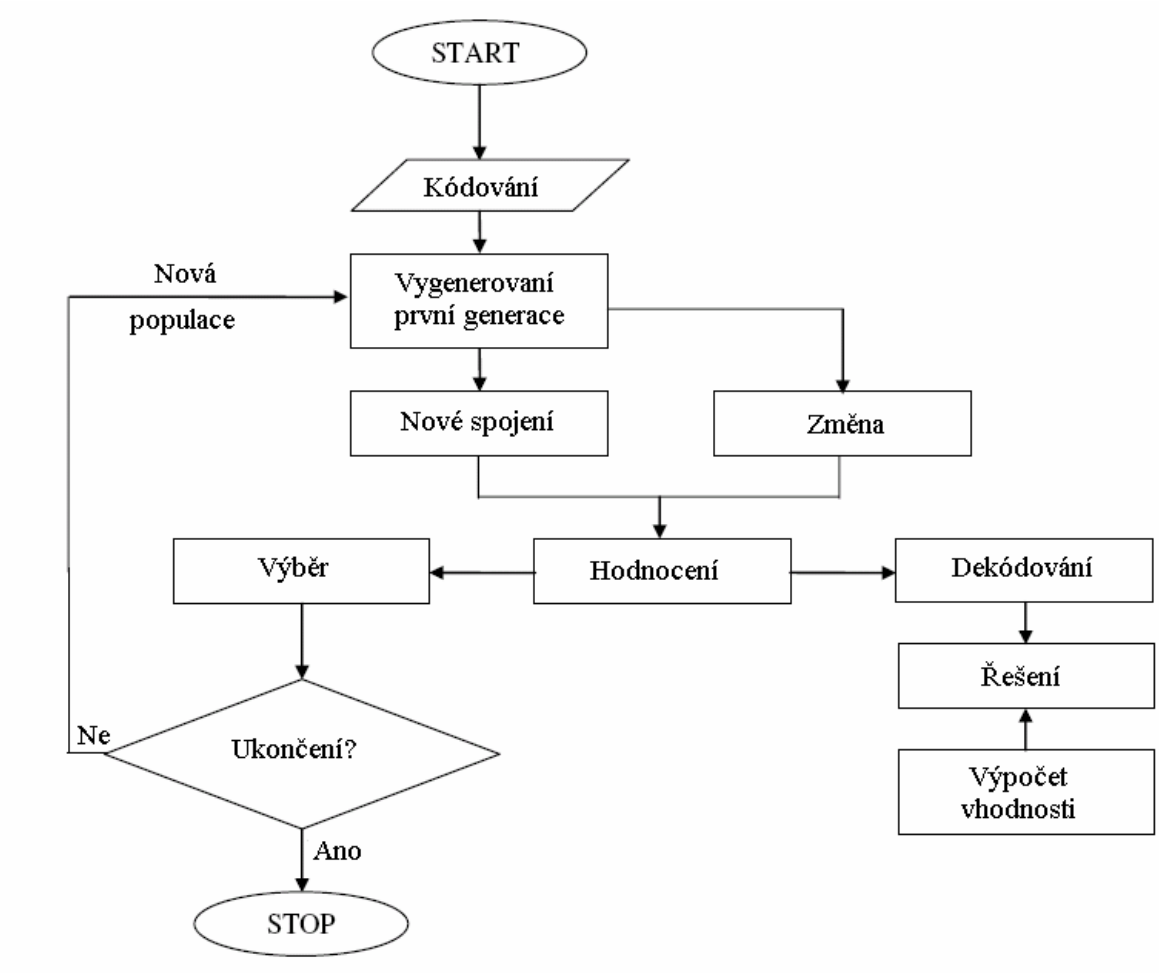
* Ing. Lucie Fišerová, Ing. Lenka Raudenská: Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav metrologie a zkušebnictví, Odbor řízení jakosti. Technická 2896/2, 616 69 Brno. tel.: 54114 2521. E-mail: yfiser04@stud.fme.vutbr.cz, yraude00@stud.fme.vutbr.cz

Inspirovány chováním těchto kolonií vznikly algoritmy tzv. Rojové inteligence, které jsou úspěšně aplikovány na řešení náročných optimalizačních úloh.

1.1 GA (Genetic algorithm)

Genetic algorithm (Genetický algoritmus) hledání založený na mechanismu přirozeného výběru a genetiky využívá klasické teorie evoluce. GA řeší nesnadné problémy rychle, spolehlivě, přesně a je široce užívaný pro řešení optimalizačních a aproximačních problémů.

Algoritmus začíná s počáteční populací (tzv. první generací) často vygenerovanou náhodně. Každý jednotlivec (nositel genetické informace) v populaci reprezentuje možné řešení problému. Populací rozumíme obecné vyjádření pro sled generací. Hodnota Fitness, síla jedince v generaci, na které závisí pravděpodobnost jeho přežití, je přidělena každému jednotlivci a provází ho celým hledáním. Vyšší hodnota Fitness dává jedinci vyšší pravděpodobnost selekce, tj. výběr jedinců, kteří přežívají v generaci. Vybraní jedinci postupují do genetického přetvoření genetickými operátory jako křížení a mutace.



Obrázek 1: Vývojový diagram znázorňující proces GA

Operátor křížení náhodně vybere dva jednotlivce jako Rodiče a vymění část jejich genetické výbavy pro vytvoření dvou nových jednotlivců. V praxi jde při klasické metodě o aritmetický průměr.

Operátor mutace pouze náhodně vybere jednoho jednotlivce z populace Rodičů a náhodně změní jeho hodnotu v chromozómu (změna konkrétního symbolu v chromozomu) a dá ho do obyvatelstva Dítě. V praxi to spočívá v přičtení náhodného vektoru ke genu (konkrétní symbol v chromozómu), přičemž onen náhodný vektor má nejčastěji normální rozdělení, tím je zaručen relativně malý posun hodnot.

Oba operátory (křížení, mutace) poměru by měly být pečlivě přizpůsobeny ke zlepšení vyhledávacího výkonu, to znamená, že každá nová generace by měla přinášet lepší řešení. Nově vygenerované obyvatelstvo - Dítě se stává obyvatelstvem Rodič pro další generaci a podstoupí stejný postup znovu.

V praxi se potom simuluje tím způsobem, že na konci každého cyklu (simulačního kroku) se zruší celá stará populace a vytvoří se nová na jejím podkladu, nebo se nechá zrušit jen část populace (nejlépe ta nejméně úspěšná) a její místo se využije k množení zbytku.

Základní kroky GA:

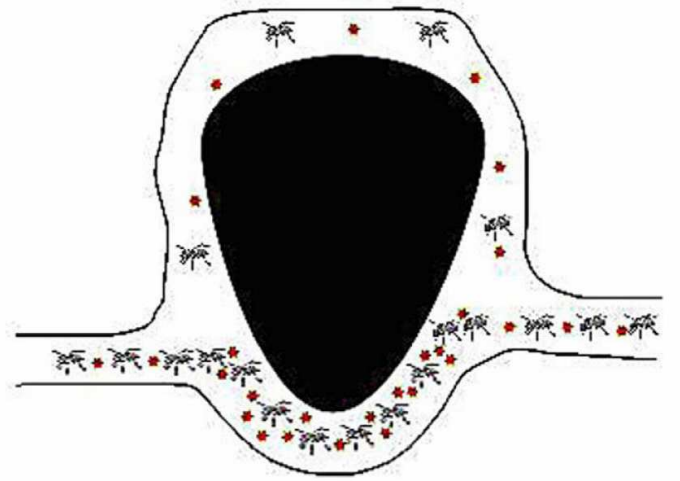
1. vygenerování počátečního obyvatelstva
2. zhodnocení Fittnes všech jednotlivců (chromozómů) v obyvatelstvu
3. opakování následujících kroků do té doby , než se dosáhne uspokojivého řešení
    a. vybrání Rodičů z obyvatelstva podle výběrového schématu
    b. vytvoření Dětí (nový jednotlivci) z vybraných Rodičů pomocí operátoru křížení
    c. mutování části jednotlivců (s nižší hodnotou vhodnosti) z obyvatelstva
    d. vytvoření dalších generací s potomstvem
    e. zhodnocení vhodnosti všech chromozómů v novém obyvatelstvu

Ukončení algoritmu je dáno dosažením uspokojivého řešení, což znamená, že se rozdíl mezi hodnotami Fittnes nejlepších jednotlivců za poslední cykly změnil jen o malou hodnotu. Nebo algoritmus zastavíme po pevně zadaném počtu iterací (bez ohledu na dosaženou přesnost).

1.2 ACO (Ant Colony Optimization)

Ant Colony Optimization (Mravenčí kolonie) je heuristická metoda inspirovaná chováním skutečných mravenčích kolonií a jejich schopností kolektivně řešit problémy. ACO se používá pro řešení nespojitého optimalizačního problému.

Systém získávání potravy v mravenčí kolonii koordinují stovky tisíc jedinců, kteří pokrývají tisíce čtverečních metrů. Mravenci při svém pohybu od hnízda ke zdroji potravy a zpět zanechávají na své cestě feromonovou stopu. V případě, že je více možných cest k potravě (viz. obrázek 2), pak každý mravenec zanechá stejné množství feromonu na každém kroku bez ohledu na zvolenou cestu (koncentrace feromonu klesá s narůstajícím časem). Mravenci zanechají vyšší koncentraci feromonu na kratší cestě, protože za stejnou dobu projdou cestu víckrát. Ostatní mravenci tak využívají míru koncentrace feromonu k určení nejkratší cesty ke zdroji potravy, to jim dává schopnost rychlejšího sběru potravy.



Obrázek 2: Feromonová stopa mravenců (naznačeno jako červené tečky) a nalezení nejkratší cesty k potravě

V praktickém využití je ACO aplikací jednoduchých pravidel, jimiž se řídí jedinci v kolonii, tedy komplexním chováním celku, který je schopný řešit složité optimalizační úlohy.

Umělí mravenci představují řešení problému a společná paměť odpovídá feromonové stopě.

1.3 PSO (Particle swarm optimization)

Particle swarm optimization (Partikulární rojová optimalizace) je jednou z posledních evolučních optimalizační technik se stochastickým přístupem, založená na chování populace.

PSO je inspirována vzájemným sociálním působením a komunikací v hejně ptáků nebo ryb. V těchto skupinách je vždy vůdce (jednotlivec s nejlepším hodnocením), který řídí pohyb celého hejna. Pohyb každého jednotlivce je založený na vůdci a na jeho vlastních znalostech. Celkově může být řečeno, že model PSO předpokládá, že chování každého jednotlivce je kompromis mezi jeho jednotlivou pamětí a soubornou pamětí.

Schéma popisuje algoritmus ve zjednodušené podobě:

1. Nastavíme počáteční hodnoty populace s náhodnými hodnotami pozice (to jest řešení) a rychlost (to jest změna vzoru/modelu řešení). Za optimální počáteční pozici je považována nejlepší pozice z celého hejna.

2. Každý jednotlivec zná svoji pozici a hodnotu účelové funkce pro tu pozici. Také si pamatuje jeho vlastní nejlepší předchozí pozici a jeho odpovídající hodnotu účelové funkce.

3. Vypočteme vhodnosti všech jedinců, a to vypočítáním hodnoty účelové funkce pro každého účastníka.

4. Porovnáme vhodnost každého jedince s jeho vlastní historickou nejlepší pozicí $p_{i,t}$. Jestliže je jeho vlastní historická nejlepší pozice $p_{i,t}$ menší pak ji nahradíme nynější hodnotou.

5. Porovnáme nejlepší momentální polohy všech jednotlivců s historickou nejlepší pozicí celého hejna $p_{\psi i,t}$. Jestliže je historická nejlepší pozice $p_{\psi i,t}$ celého hejna menší, pak ji nahradíme momentální polohou všech jednotlivců.

6. Obnovíme pozice a rychlosti všech jednotlivců dle níže uvedených rovnic.

$$v_{i,t+1} = c_1 v_{i,t} + c_2(p_{i,t} - x_{i,t}) + c_3(p_{\psi i,t} + x_{i,t})$$

$$x_{i,t+1} = x_{i,t} + v_{i,t+1}$$

Proměnné:

$x_{i,t}$ ... pozice jednotlivce *i* v iteraci *t* (ekvivalent jednoho řešení problému)

$v_{i,t}$ ... rychlost jednotlivce *i* v iteraci *t* (ekvivalent ke změně vzoru řešení)

$p_{i,t}$ ... nejlepší předchozí (myslí se tím nejlepší pozice ze všech předchozích $x_{i,t}$ – nejlepší se pozná dle hodnoty účelové funkce) pozice jednotlivce *i* v iteraci *t* (zapamatovaná každým jednotlivcem)

7. Posoudíme, zda je kritérium pro ukončení naplněno, jestliže "ano" potom ukončíme iterace; jestliže "ne" vrátíme se ke kroku (3). Kritérium pro ukončení může být zadáno jako počet iterací, hodnota vhodnosti jednotlivců či nemožností dosažení optimálního řešení.

PSO má širokou použitelnost, většina z jeho aplikací se soustřeďuje na optimalizaci funkcí, je taktéž aplikován na kombinatorické optimalizační problémy.

1.4 The Bees Algorithm

The Bees Algorithm (Včelí algoritmus) je založen na chování populace, napodobuje roj včel při sběru pylu. Ve svém základě algoritmus představuje hledání v okolí kombinovaný s náhodným hledáním a může být použit jak pro kombinatorickou, tak pro funkční optimalizaci.

Včelstvo je schopno se za účelem sbírání pylu rozprostřít různými směry do velmi velkých vzdáleností (až 10 km), a tím využít více zdrojů potravy. Na základě rozmístění včel dělnic se včelstvu daří soustředit na kvalitní pylové oblasti. Základním principem je myšlenka, že úrodná pole s množstvím pylu, který může být nasbírán s menším úsilím, jsou navštěvována více včelami, zatímco místa s méně pylem navštíví méně včel.

Proces začíná vysláním nejzdatnějších včel k prohledání okolí. Včely se při hledání pohybují náhodně. Po návratu do úlu vyjadřují tancem tři základní údaje: kvalitu, směr a vzdálenost pole. Všechny znalosti o vnějším prostředí pocházejí pouze z tohoto tance, který současně umožňuje včelstvu ohodnotit vhodnost různých míst podle kvality pylu a energie potřebné ke sběru. Poté letí včely průzkumnice na pole společně se včelami dělnicemi, které doposud čekaly v úlu. Více včel je vysláno na místa s předpokládanou vyšší kvalitou. Toto umožňuje včelstvu shromáždit pyl rychle a efektivně.

Proces prozkoumávání terénu se stále opakuje, takže informace o místech jsou stále aktuální a včely jsou dle nich vysílány na původní případně i na úplně nová místa.

Schéma zjednodušeně popisuje algoritmus:

1. definice místa k optimalizaci

2. definice populace (počtu vzorků)

3. vybrání nejkvalitnějších zástupců z populace

4. náhodné rozmístění zástupců

5. sběr a vyhodnocení nasbíraných informací

6. vyslání populace na nejkvalitnější místa

V praxi začíná algoritmus definováním místa k optimalizaci a vygenerováním populace, z které se následně vyberou nejkvalitnější zástupci – tzv. průzkumníci. Průzkumníci jsou umístěni náhodně ve zkoumaném prostoru a je jimi ohodnocena vhodnost navštívených míst. Poté následuje celkové posouzení a průzkum se soustředí především na nejperspektivnější oblasti. Tento krok se opakuje, dokud není dosaženo dané kvalitativní kritérium.

Exaktní popis Včelího algoritmu:

Algoritmus vyžaduje zadání následujících proměnných: *n, m, e, ngh, nep* a *nsp,* spolu s kritériem pro ukončení.

Proměnné:

n.…počet průzkumníků

m.…počet vybraných stran z n možných stran

e.…počet nejlepších stran z *m* stran

ngh…počáteční velikost místa

nep.…počet včel v okolí nejlepšího bodu

nsp.…počet včel v okolí

Samotný algoritmus začíná s *n* průzkumníky, kteří jsou náhodně umístěni do zkoumaného prostoru. Hodnota vhodnosti navštívených stran je určena v kroku 2.

1. vytvoření populace s náhodnými hodnotami
2. ohodnocení vhodnosti populace
3. vybrání směrů pro průzkum
4. poslání průzkumníků na vybrané strany (více průzkumníků na nejlepších *e* stran) a ohodnocení vhodnosti
5. vybrání průzkumníků s největší hodnotou vhodnosti
6. přiřazení zbývajících průzkumníků k náhodnému vyhledávání a ohodnocení jejich vhodnosti
7. v případě, že není naplněno kritérium zastavení, vytvoření nové populace a návrat na začátek algoritmu

V kroku č. 3 jsou vybráni průzkumníci s nejvyšší hodnotou vhodnosti jako „výběrové včely“ a směry jimi navštívené jsou vybrány pro podrobnější prozkoumání jejich okolí. V čtvrtém a pátém kroku algoritmu se provádí hledání v okolí vybraných stran a přiřazuje se větší množství průzkumníků pro hledání v nejlepších místech *e.* Směry, které včely navštíví, mohou být vybrané dle hodnoty vhodnosti. Eventuelně je tato hodnota využita pro určení pravděpodobnosti, že bude průzkumník vybrán. Do blízkosti nejlepšího směru (směr slibující nejlepší výsledky) je vysláno více průzkumníků oproti ostatním oblastem, což přestavuje klíčovou operaci Včelího algoritmu.

V kroku č. 5 je pro každé místo vybrán pouze jeden průzkumník s vysokou hodnotou vhodnosti k vytvoření další populace. V přírodě ovšem takovéto pravidlo není. Toto pravidlo slouží v algoritmu ke snížení počtu míst ke zkoumání. V kroku č. 6 jsou zbývající průzkumníci náhodně přiřazeni k novým místům.

Na konci každé iterace má roj dvě části populace – jednotlivce z vybraných míst a další průzkumníky určené k hledání.

Tyto kroky jsou opakovány, dokud není naplněno kritérium pro ukončení algoritmu.

2 Porovnání Swarm-based optimisation algorithms

Předchozí algoritmy byly použity v šesti benchmarkingových funkcích a výsledky zaneseny do tabulky. Počet iterací v tabulce byl získán jako průměr 100 nezávislých měření.

Tabulka: Srovnání metod

Název funkce	GA		ACO		Včelí Algoritmus	
	Počet iterací	Úspěšnost [%]	Počet iterací	Úspěšnost [%]	Počet iterací	Úspěšnost [%]
De Jong	10160	100	6000	100	49	100
Goldstein & Price	5662	100	5330	100	999	100
Branin	7325	100	1936	100	1657	100
Martin & Gaddy	2844	100	1688	100	526	100
Rosenblock	10212	100	6842	100	898	100
Hyper sphere	15468	100	22050	100	7113	100
Griewangk	200000	100	50000	100	1847	100

První je De Jongova funkce, pro kterou našel Včelí Algoritmus optimum stodvacetkrát rychleji než ACO a 207 rychleji než GA s úspěšností 100%. Druhou funkcí je Goldstein & Price, kde Včelí algoritmus dosáhl optimum pětkrát rychleji než ACO a GA taktéž se 100% úspěšností. U Braninovy funkce je Včelí algoritmus o 15% rychlejší ve srovnání s ACO a o 77% rychlejší oproti GA taktéž se 100% úspěšností. U modelu Hyper Sphere, který je šesti-dimenzionální, potřeboval Včelí algoritmus o polovinu méně iteraci oproti GA a pouze třetinu oproti ACO. Poslední funkce Griewangk je deseti-dimenzionální, kde Včelí algoritmus dosáhl optima desetkrát rychleji než GA a pětadvacetkrát rychleji než ACO, opět s úspěšností 100%.

3 Závěr

V současnosti jsou pro řešení problémů, kde není znám exaktní algoritmus pro nalezení optima účelové funkce, převážně používány algoritmy rojové inteligence. Nejefektivnější algoritmus je takový, který dosahuje uspokojivého výsledku v nejmenším počtu iterací. Podle

benchmarkingových funkcí je nejefektivnější Včelí algoritmus, který je inspirován chováním včel ve volné přírodě při sběru pylu.

V praxi jsou jednotlivé metody využívány v různých případech. Například Ant Colony Optimization je s úspěchem využívána na řešení problému obchodního cestujícího a Genetický algoritmus na VRP (Vehicle Routing Problem).

V budoucnosti by bylo velmi přínosné zaměřit se na zlepšování systémů a procesů v podnicích pomocí výše popsaných optimalizačních metod. Toto zlepšování povede jak k zvýšení kvality, tak i produktivity celého provozu.

Seznam použité literatury

D. T. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, M. Zaidi. *The Bees Algorithm – A Novel Tool for Complex Optimisation Problems*. Manufacturing Engineering Centre, Cardiff University.

M. Izadifar, M. Zolghadri Jahromi. *Application of genetic algorithm for optimization of vegetable oil hydrogenation process*. 2004. Journal of Food Engineering 78 (2007) 1–8

H. Stern, Y. Chassidim, M. Zofi. *Multiagent visual area coverage using a new genetic algorithm selection scheme*. 2003. European Journal of Operational Research 175 (2006) 1890–1907

A. Kumar, Prakash, M.K. Tiwari, R. Shankar, A. Baveja. *Solving machine-loading problem of a flexible manufacturing system with constraint-based genetic algorithm*. 2004. European Journal of Operational Research 175 (2006) 1043–1069

Erick Cantú-Paz, D. E. Goldberg. *Effcient parallel genetic algorithms: theory and practice*. 1999. Comput. Methods Appl. Mech. Engrg. 186 (2000) 221±238

A. Kumar. European Journal of Operational Research 175 (2006) 1043-1069

Genetické algoritmy. <http://www.sweb.cz/labirlab/algoritmy/genetic.htm>

Seznámení se s genetickými algoritmy. <http://mujweb.cz/www/prikrylj/Genetickealgoritmy.html#1>

A. Bahreininejad, P. Hesamfar. *Subdomain generation using emergent ant colony optimization.* 2005. Computers and Structures 84 (2006) 1719–1728

Optimalizace pomocí mravenčích kolonií <http://www.milosnemec.cz/clanek.php?78>

Ant Colony Optimization. <http://aco.wz.cz/algoritmy.php>

C. Andrés, S. Lozanob. *A particle swarm optimization algorithm for part–machine grouping*. 2005. Robotics and Computer-Integrated Manufacturing 22 (2006) 468–474

Sheng-Fa Yuana, Fu-Lei Chua. *Fault diagnostics based on particle swarm optimisation and support vector machines*. 2006. Mechanical Systems and Signal Processing

Qie He, Ling Wang, Bo Liu. *Parameter estimation for chaotic systems by particle swarm optimization*. 2006. Chaos, Solitons and Fractals (2006)