# Agent Collaboration for Multiple Trading Strategy Integration⋆

Longbing Cao[1], Dan Luo[2], Yanshan Xiao[3], and Zhigang Zheng[4]

[1,2] Faculty of Information Technology, University of Technology, Sydney, Australia
[3] Department of Information Technology, Guangzhou Asian Games Organizing Committee, China
[4] Department of Information Technology, PICC, Property And Casualty Ltd, China
lbcao@it.uts.edu.au

**Abstract.** The collaboration of agents can undertake complicated tasks that cannot be handled well by a single agent. This is even true for excecuting multiple goals at the same time. In this paper, we demonstrate the use of trading agent collaboration in integrating multiple trading strategies. Trading agents are used for developing quality trading strategies to support smart actions in the market. *Evolutionary* trading agents are armed with evolutionary computing capability to optimize strategy parameters. To develop even smarter trading strategies (we call *golden strategies*), multiple *Evolutionary* and *Collaborative* trading agents negotiate with each other for $m$ loops to search multiple local strategies with best parameter combinations. They also integrate multiple classes of strategies for trading agents to achieve the best global objectives acceptable for trader needs. Tests of five classes of trading strategies in ten years of five markets of data have shown that agent collaboration for strategy integration can achieve much better performance of trading compared with that of either individually optimized or randomly chosen strategies.

## 1 Introduction

Collaboration, coordination, cooperation and negotiation are some of key organizational activities in a multi-agent system. Agent collaboration can undertake complicated tasks that usually cannot be handled well by a single agent. This is even true for executing multiple goals at the same time. Another key concept used in this paper is trading agent. Trading agent [5,6,13,14] is a concept developed to design and simulate market mechanisms, auction strategies, and supply chain management etc. The collaboration of trading agents is interesting because in this way it may develop trading support information that cannot be achieved by single agents.

This paper demonstrates the use of trading agent collaboration for integrating multiple classes of trading strategies to support smart trading. We develop

⋆ This work is sponsored by Australian Research Council Discovery Grant (DP0773412, LP0775041, DP0667060), and UTS internal grants.

N.T. Nguyen et al. (Eds.): KES-AMSTA 2008, LNAI 4953, pp. 361–370, 2008.

*Evolutionary* trading agents through arming agents with evolutionary computing capabilities. Evolutionary trading agents are therefore able to identify most interesting strategies for smart trading. Further, *Collaborative* agents negotiate with *Representative* trading agents for $m$ loops if necessary to search for the locally optimal trading strategies and then aggregate some of these locally optimal strategies to generate a global optimization objective.

In developing the integrative trading agents, our concern is *to what extent the trading agents powered by the existing approaches can support both technical significance and business decision making [11,8] in real-life marketplaces.* To this end, we customize trading agents with the tasks of satisfying both *trader preference* and taking trading positions in favor of trader's expectation. In general, there are three types of positions: +1, −1, 0 for a trading agent to take in the market. Position +1 indicates a *buy* or *holding buy* action in the market. Position −1 reflects either a *sell* or *holding sell* action. Position 0 indicates none of actions.

Any positions undertaken by a trading agent are associated with certain benefit and cost/risk. Trading agents need to carefully select those trading strategies that can guide them to take positions in the market with higher benefits while controlling costs. To balance benefit and cost, the aim of trading agents is to maintain the highest unit of benefit per cost $\gamma_{\alpha\beta}$.

**Definition 1 (Benefit-Cost Ratio).** Benefit-cost ratio $\gamma_{\alpha\beta,s}$ *measures the the unit of benefit per cost of a trading agent in undertaking position sequences* $\{b_i\}$ *determined by a trading strategy* $s$.

$$\gamma_{\alpha\beta,s} = \alpha_s/\beta_s \tag{1}$$

We use stock trading agents to illustrate the development of actionable trading strategies and multi-strategy integration. Evolutionary trading agents for optimizing parameters and trading agent collaboration for strategy integration are proposed to develop smart trading agents for actionable strategies. Seven categories including 36 types of trading strategies have been developed for stock trading agents. Ten years of historical data from five markets are used to evaluate the actionability of the stock trading agents. Massive experiments have shown that our trading agents present excellent performance that can not only beat naive strategies, but also financial market benchmarks.

## 2 Designing Smart Trading Agents

The idea of *designing smart trading agents* is to endow trading agents with capabilities of searching strategies in constrained market environment to satisfy trader preference. In this section, we introduce two approaches to designing smart trading agents. One is to design *evolutionary trading agents*, which are equipped with evolutionary computing capabilities, and can search strategies from a large candidate strategy space targeting higher *benefit-cost ratio*. The other is to integrate [7] optimal instances from multiple classes of trading strategies into one combined powerful strategy through *collaborative trading agents*.

### 2.1 Evolutionary Trading Agents for Parameter Optimization

*Evolutionary trading agents* have capabilities of evolutionary search computing. They can search trading strategies based on given optimization fitness and specified optimization objectives. Their roles consist of *optimization requests* (including base strategies and arguments), *creating strategy candidates* (namely chromosomes), *evaluating strategy candidates*, *crossing over candidate strategies*, *mutating candidate strategies*, *re-evaluating candiate strategies*, and *filtering optimal strategies*, etc.

The strategy optimization mechanism of evolutionary trading agents is as follows. A *User Agent* receives optimization requests from user-agent interaction interfaces. It forwards the request to *Coordinator Agents*, *Coordinator Agents* check the availability and validity of optimized *Strategy Agent* class with *strategyClassID*. If a *Strategy Agent* class is available and optimizable, *Coordinator Agents* call the *Evolutionary Agents* to perform corresponding roles, for instance, *createStrategyCandidates*, *evaluateStrategyCandidates*, *crossoverCandidateStrategies*, *mutateCandidateStrategies*, *re-evaluateCandidateStrategies*, or *returnOptimalStrategies* to optimize the strategy. After the optimization process, *Evolutionary Agents* return *Coordinator Agents* the searched optimal *Strategy Agent* with *strategyID* and corresponding parameter values. *Coordinator Agents* further call the *User Agents* to present the results to traders by invoking *Presentation Agents*. Fig. 1 illustrates the workflow of the above evolutionary trading agents and their relevant collaboration process in searching actionable trading strategies.

The following descriptive notations further illustrate one of the above roles: *mutateCandidateStrategies*.

Role *R_mutateCandidateStrategies*
Statement *Mutation is a process that parts of a chromosome are to be changed. This role determines to what extent the parts of a chromosome in a trading agent are to be mutated. The extent is the mutation rate.*
Agent *A_EvolutionaryAgent*
Agent *A_UserAgent*
Agent *A_StrategyAgent*
Agent *A_CoordinatorAgent*
Attribute *aea:A_EvolutionaryAgent*
Attribute *constant mutrate:MutationRate*
Attribute *paraid[]:A_InParameters*
Attribute *aua:A_UserAgent*
Attribute *asa:A_StrategyAgent*
Attribute *constant strid:asa*
Attribute *aca:A_CoordinatorAgent*
Protocol *receiveStrategyMutationRequest*
Protocol *checkStrategyAgentValidity*
Protocol *openMutateSettingInterface*
Protocol *submitStrategyMutationRequest*

Protocol *returnStrategyMutationResponse*
Responsibilities
Liveness
$\forall$ *strid:aca.checkStrategyAgentValidity()* $\rightarrow$
*aua.openMutateSettingInterface(aea, asa.paraid[])*
$\rightarrow$ *aea.receiveStrategyMutationRequest(aua)*
$\rightarrow$ *aca.submitStrategyMutationRequest(aua)*
$\rightarrow$ *aea.executeStrategyMutation(aua, mutrate, aca)*
$\rightarrow \Diamond_{\leqslant t}$ *aea.returnStrategyMutationResponse(aua, aca)*
Safety (Invariant) $0 < mutrate < 1.0$

### 2.2 Trading Agent Collaboration for Strategy Integration

In real-life trading, trading strategies can be categorized into many classes. To financial experts, different classes of trading strategies indicate varying fundamental principles of market models and mechanisms. A trading agent often takes series of positions generated by a specific trading strategy, which instantiates a trading strategy class. Trading agents can collaborate to take concurrent positions created by multiple trading strategies to take advantage of varying strategies.

The idea of *trading agent collaboration for strategy integration [7]* is as follows. There are a few *Representive Trading Agents* in the market. Each *Representive Agent* invokes an *Evolutionary Agent* to search for optimal *Strategy Agent* from a strategy class. *Coordinator Agents* then negotiate with these *Representive Agents* and *Evolutionary Agents* to integrate the identified optimal strategies of *Strategy Agents*. An appropriate integration method is negotiated and chosen by *Coordinator Agents*, *Representive Agents* and *Evolutionary Agents* based on globally optimal output.

For instance, the following notations describe one of the goals of *Coordinator Agents*. The goal is to achieve the globally maximal *benefit-cost ratio* through negotiating with all *Representive Agents*.

Goal *integrateStrategy*
Statement Coordinator *agents discuss with* Representative *trading agents to get maximally* global benefit-cost ratio. Representative *trading agents invoke* n Evolutionary *trading agents to execute* n *classes of* Strategy *agents for maximally* local benefit-cost ratio, *respectively. The following describes the objective of agents fulfilling such a task.*
Role *R_StrategyOptimizer*
Agent *A_StrategyAgent*
Agent *A_UserAgent*
Agent *A_RepresentativeAgent*
Agent *A_EvolutionaryAgent*
Agent *A_CoordinatorAgent*
Attribute $aea$ : $A\_EvolutionaryAgent_i$
Attribute $aua$ : $A\_UserAgent$

Attribute $asa$ : $A\_StrategyAgent_i$
Attribute $ara$ : $A\_RepresentativeAgent$
Attribute $aca$ : $A\_CoordinatorAgent$
Attribute $constantstrid$ : $asa$
Attribute $constantstrid$ : $asa$
Attribute $an$ : $AlgoName$
Attribute $ac$ : $AlgoCode$
Attribute $ain[]$ : $AlgoParameters$
Attribute $aout[]$ : $AlgoOutputs$
Creation condition $\neg Existed(ac)$
Invariant condition $ac.actor = ActorID$
Fulfillment condition
$\forall$ *ac:AlgorithmComponent (*
*ac.algo = algo* $\rightarrow$
$\Diamond_{\leqslant t_1} \exists$ *cpi:CallPluginInterfaces (cpi.actor = actor* $\wedge$ *Fulfilled(cpi))*
$\wedge\ \Diamond_{\leqslant t_2}(\exists$ *faro:FillinAlgoRegisterOntologies*
*(faro.depender = actor* $\wedge$ *Fulfilled(faro))*
$\wedge\ \exists$ *uac:UploadAlgoComponent*
*(uac.depender = actor* $\wedge$ *Fulfilled(uac)* $\wedge$ *ac.uploaded)*
*)*
*)*

Fig. 1 further describes the process of trading agent negotiation for strategy integration. As shown in the diagram, there are two steps of optimzation. First, locally optimal strategies are searched through *Evolutionary* agents on request of *Representative Agents* if the strategy achieves the highest *benefit-cost ratio* $\sigma$. *StrategyManager Agent* stores the golden strategies. Second, *Coordinator* agents call *StrategyIntegrator Agents* to work out the requested global goal. *Coordinator* agents check *StrategyManager Agent* and invoke *Evolutionary* agents if necessary to recalculate the golden strategies based on negotiation model. *StrategyIntegrator Agents* select the best golden strategies for each loop and accumulate all promising strategies for m loops to achieve the requested globally optimal goal by following the agreed negotiation model.

## 3 Real-World Experiments

Since 2002, we have been working on developing trading agents and strategies with industrial partners' support, say CMCRC, SIRCA and SMARTS [1,3,2]. Massive experiments have been conducted on many years of multi-markets of data. An agent service-based platform F-Trade [4,9] has been built to support this effort. Some of our results have been delivered to partners. In this section, we illustrate the process and results in optimizing strategies through *Evolutionary Trading Agents* and integrating strategies via *Collaborative Trading Agents*.

Given a trading strategy $s$, a trading strategy class $S_i$ ($i$=1, 2, ...), $s \in S_i$, $\alpha_s$ and $\beta_s$ are the $benefit$ and *cost* of a trading agent in executing the

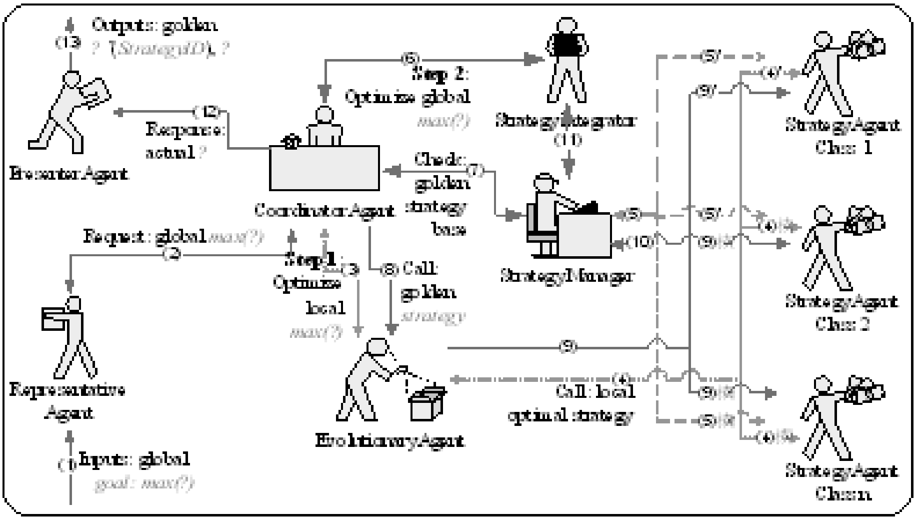


**Fig. 1.** Workflow of trading agent collaboration for strategy integration

strategy $s$. The development process of integrating strategies through trading agent collaboration is as follows.

Part A. *Data Manager Agent* prepares data:
0). *UserAgent* receives trader's input requests;
1). *DataManager* agent splits two years of data for training;
2). *RepresentativeAgent* invokes *EvolutionaryAgents* to identify locally golden trading strategies with highest $\gamma_{\alpha\beta,s}$ as discussed in part B;
3). *DataManager* agent splits another three years of data following the training windows for testing;
4). *RepresentativeAgent* invokes *EvolutionaryAgents* to test the identified golden strategies as discussed in part B;
5). *DataManager* agent slides the 2-year training and the 3-year deploying data windows one year forward to extract data sets as in A:1) and A:3);
6). *RepresentativeAgent* invokes *EvolutionaryAgents* to repeat the operations of searching golden strategies;
Part B. *Evolutionary Trading Agents* searching golden strategies:
1). *EvolutionaryAgent* calls a *StrategyAgent* $s$ in class $S_i$ and searches strategy instance $s'$ with $max(\alpha_{s'})$ for $s'$ positions;
2). *EvolutionaryAgent* calls a *StrategyAgent* $s$ and searches strategy $s''$ with $max(\gamma_{\alpha\beta,s})$ when $s''$ positions are executed;
3). *RepresentativeAgent* invokes *EvolutionaryAgents* to search all strategies $s''_i$ $((i = 1, 2, \ldots))$ in all strategy classes satisfying conditions in step B:2) respectively;
Part C. *Collaborative Trading Agents* aggregate golden strategies:
1). *PositionAgents* extract all positions from *EvolutionaryAgents* with all

strategies identified in step B:3) for *RepresentativeAgent*;
2). *EvaluationAgents* check the *benefits*, *costs* and *benefit-cost ratio* of each *RepresentativeAgent* executing the above positions;
3). *DecisionAgents* filter out strategies with low $\gamma_{\alpha\beta,s}$ for each strategy class $i$;
4). *CoordinatorAgents* call all *RepresentativeAgents* to execute the above filtered strategies concurrently to generate the final outcomes.

Experiments of trading agent collaboration for multi-strategy integration in stock market data have been conducted as follows:

- Five classes of trading strategies are developed: MA, FR, CB, SR, and OBV;
- Five stock markets: ASX, Hongkong, London, New York, and Japan;
- Interday data from 1/11/1998 to 31/10/2007: date, price, volume as shown in Table 1;
- Training data: 2-year sliding window, say 1/11/1998-31/10/1999;
- Testing data: 1-year sliding window, say 1/11/1999-31/10/2000.

**Table 1.** Data sample

| Date | Price | Volume |
|---|---|---|
| 2006-12-14 | 16.39 | 239943 |
| 2006-12-15 | 16.74 | 183908 |
| 2006-12-18 | 17.25 | 203883 |
| 2006-12-19 | 16.97 | 178483 |

Fig. 2 illustrates some results of evolutionary trading agent for optimizating the *Filter Rule Base Strategy* $FR(x)$. $FR(x)$ indicates a generic class of correlated trading strategies, by which you go long at the time that the price rises by x% and hold until the price falls x%, at which time you close out and go short, where $x \in [0, 1]$ is the percentage price movement of highest high and lowest low.

Even though there is only one parameter d in this rule, it is hard to find the most appropriated 'x' in real-life market. Evolutionary trading agent is helpful for searching the golden 'x'. As shown in Fig. 2, the cumulative payoff with $x = 0.04$ always outperform other ds from 14 July 2003 in trading the listed security CBA (Australian Commonwealth Bank) in Australian Stock Exchange (ASX) in 2003-2004.

Table 2 shows the signals, positions, benefits and costs of trading agents following *MA-BMN Strategy*, which is an identified golden strategy by evolutionary trading agent in 2004 Hongkong Exchange data.

Table 3 shows the positions recommended by each golden strategy identified by collaborative trading agents in 2006 Hongkong Exchange data.

Fig. 3 shows the cumulative benefits *MA_Ben, FR_Ben, CB_Ben, SR_Ben, OBV_Ben* of trading agents taking positions recommended by golden trading strategies *MA, FR, CB, SR, OBV*, as well as that (Int_Ben) of the *Collaborative*

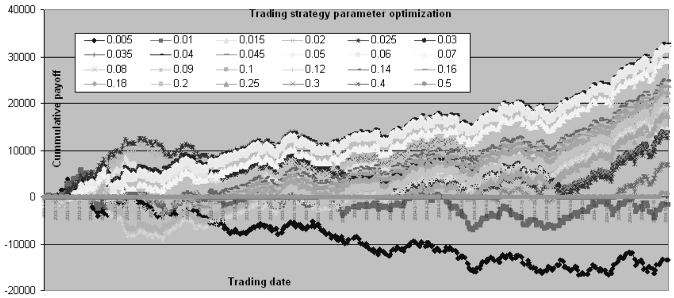


**Fig. 2.** Some results of evolutionary trading agent for strategy optimization

**Table 2.** Output excerpt of a trading strategy

| Date | Price | Sell | Buy | Position | Benefit($) | Cost($) |
|---|---|---|---|---|---|---|
| $2004-8-16$ | 3466 | $-1$ | 0 | $-1$ | 9200 | 103 |
| $2004-8-17$ | 3480 | $-1$ | 0 | $-1$ | 8850 | 106.5 |
| $2004-8-18$ | 3472 | $-1$ | 0 | $-1$ | 9150 | 108.5 |
| $2004-8-19$ | 3481 | $-1$ | 0 | $-1$ | 8825 | 110.75 |
| $2004-8-20$ | 3494 | 0 | 0 | $-1$ | 8500 | 114 |

**Table 3.** Trading agent positions recommended by five trading strategy classes (excerpt)

| Date | MA Pos | FR Pos | CB Pos | SR Pos | OBV Pos |
|---|---|---|---|---|---|
| $2006-11-16$ | 1 | 1 | 0 | 1 | 1 |
| $2006-11-17$ | 1 | 1 | 0 | 1 | 1 |
| $2006-11-20$ | 1 | 1 | 0 | 1 | 1 |
| $2006-11-21$ | $-1$ | $-1$ | 0 | 1 | 1 |
| $2006-11-22$ | $-1$ | $-1$ | 0 | 1 | 1 |

*Trading Agent* executing all golden positions by integrating individual golden strategies concurrently in 2003-2006 Hongkong United Exchange data.

A large amount of tests in stock data of five markets have shown trading agents following all golden trading strategies can obtain higher benefits and benefit/cost ratios (except FR in the first few days). In particular, collaborative trading agents concurrently executing positions from all individual golden strategies can greatly increase benefits while control very low costs compared with those taking positions recommended by either an individually golden strategy or randomly chosen strategies only (see Table 4, *lift* [10] measures how much good a trading strategy is in all split data sets).

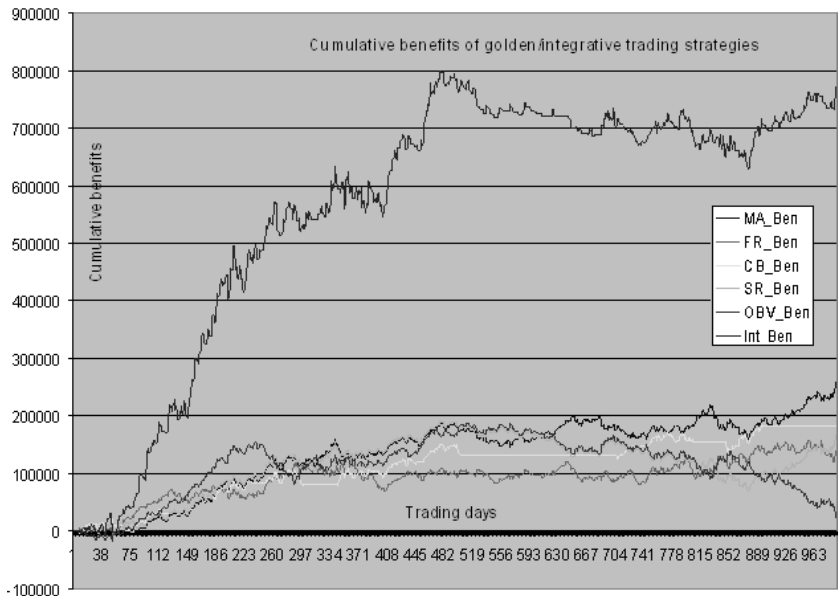


**Fig. 3.** Cumulative benefits of trading agents following golden trading strategies

**Table 4.** Lift comparison between random chosen strategies and golden strategies

| Lift | MA-CMN | FR-XY | OBV-B | CB-XNC | SR-NC |
|---|---|---|---|---|---|
| Random | 10% | 0 | 20% | 10% | 10% |
| Optimized | 70% | 80% | 80% | 90% | 100% |

## 4 Conclusion

Agent collaboration and negotiation is very helpful for solving complicated tasks and achieve multiple goals at the same time. Trading agent has demonstrated its potential in simulating market mechanism design and strategy development. This paper has demonstrated the use of agent collaboration for optimizing and aggregating multiple classes of trading strategies. Trading agent can contribute to traders with trading strategies that can support their action-taken in the market. First of all, trading agents are armed with evolutionary computing capability. The evolutionary capability enables trading agents to search for parameter combinations with the most appropriate performance. Further, evolutionary and collaborative trading agents collaborate with each other to generate locally optimal trading strategies, and then produce integrative and globally optimal strategies. The integrative trading strategies enable trading agents with trading positions that can lead to higher profit but lower costs.

## References


1. CMCRC: www.cmcrc.com
2. SIRCA: www.sirca.org.au
3. SMARTS: www.smarts.com.au
4. F-Trade: www.f-trade.info
5. Trading Agent Competition: http://www.sics.se/tac/
6. Trading Agent Design and Analysis: http://www.sics.se/tada05/
7. Cao, L.B.: Multi-strategy integration for actionable trading agents. In: Proceedings of the International Conference on Intelligent Agent Technology, IEEE Computer Society Press, Los Alamitos (2007)
8. Cao, L.B., Zhang, C.Q.: Two-Way Significance of Knowledge Actionability. Int. J. of Business Intelligence and Data Mining (2007)
9. Cao, L.B., Zhang, C.Q.: F-Trade: An Agent-Mining Symbiont for Financial Services. In: Proceedings of the International Conference on Autonomous Agents and Multiagent Systems, ACM Press, New York (2007)
10. Cao, L.B., Luo, C., Zhang, C.Q.: Developing Actionable Trading Strategies for Trading Agents. In: Proceedings of the International Conference on Intelligent Agent Technology (2007)
11. Cao, L.B., Luo, D., Zhang, C.Q.: Fuzzy Genetic Algorithms for Pairs Mining. In: Yang, Q., Webb, G. (eds.) PRICAI 2006. LNCS (LNAI), vol. 4099, pp. 711–720. Springer, Heidelberg (2006)
12. Chan, T.: Artificial Markets and Intelligent Agents, PhD thesis, Massachusetts Institute of Technology (2001)
13. Wan, H.A., Hunter, A.: On Artificial Adaptive Agents Models of Stock Markets. Simulation 68, 279–289 (1997)
14. Wellman, M.P., et al.: Designing the Market Game for a Trading Agent Competition. IEEE Internet Computing 5(2), 43–51 (2001)