Knowl Inf Syst (2009) 18:183–198
DOI 10.1007/s10115-008-0170-2

REGULAR PAPER

Developing actionable trading agents

Longbing Cao · Tony He

Received: 18 October 2007 / Revised: 9 March 2008 / Accepted: 16 May 2008 /
Published online: 10 October 2008

Abstract Trading agents are useful for developing and back-testing quality trading strategies to support smart trading actions in the market. However, most of the existing trading agent research oversimplifies trading strategies, and focuses on simulated ones. As a result, there exists a big gap between the deliverables and business needs when the developed strategies are deployed into the real life. Therefore, the actionable capability of developed trading agents is often very limited. This paper for the first time introduces effective approaches for optimizing and integrating multiple classes of strategies through trading agent collaboration. An integration and optimization approach is proposed to identify optimal trading strategy in each category, and further integrate optimal strategies crossing classes. Positions associated with these optimal strategies are recommended for trading agents to take actions in the market. Extensive experiments on a large quantity of real-life market data show that trading agents following the recommended strategies have great potential to obtain high benefits while low costs. This verifies that it is promising to develop trading agents toward workable and satisfying business needs.

Keywords Trading agent · Trading strategy · Optimization · Integration

1 Introduction

Agent-based computing has been introduced into trading simulation and development, for instance, trading agent [5,6,22,23]. Trading agent is a concept developed to design and

This work is sponsored in part by Australian Research Council Grants (DP0773412, LP0775041, DP0667060).

L. Cao (✉)
Faculty of Engineering and Information Technology, University of Technology Sydney,
Sydney, NSW, Australia
e-mail: longbing.cao-1@uts.edu.au

T. He
School of Finance and Economics, University of Technology Sydney, Sydney, NSW, Australia

simulate market mechanisms, auction strategies, and supply chain management etc. For instance, competitions and simulations [3,5,23] have been developed in terms of travel agents, supply chain management agents and market design.

Most of the present trading agent research either focuses only on research problems designed for specific artificial marketplaces and simulated data, or oversimplifies the market environment where trading agents live. This leads to a typical issue that the real market can benefit little from the developed strategies and trading agents. This likely results from the fact that *market organizational factors* and *business preferences* have not been sufficiently considered in the existing trading agent research. As a result, this has slowed down and even affected the roles of trading agent techniques in boosting real-life market development. It leads to our concern that is *how trading agents can support the development of innovative and workable marketplaces and business decision making as highly expected by business people* [15].

This paper specifically addresses the above issue, namely developing actionable trading agents that can produce optimal trading strategies to be directly used by business users to take decision-making actions in real-life market situations. The principal is as follows. We first define the problem of searching actionable trading strategies for trading agents. Market organizational factors and trader preferences are then discussed in enhancing strategy actionability. Further, we develop evolutionary trading agents to optimize parameters, and trading agent collaboration to integrate strategies for smart trading.

We use stock trading agents to illustrate the development of actionable trading agents and multi-strategy integration through agent collaboration. Extensive experiments have been conducted in developing hundreds of trading strategies in large amounts of multiple market data. It shows that our trading agents are able to lead to trading performance that beats not only naive strategies but also financial market benchmarks.

The remainder is organized as follows. Section 2 defines the problem of searching actionable strategies for trading agents. In Sect. 3, business factors and expectations are discussed to strengthen the actionability of trading agents. In Sect. 4, two types of smart trading agents, namely evolutionary and collaborative trading agents are developed to identify actionable trading strategies in market situations. Section 5 illustrates the use and performance of the above trading agents in stock market data. We conclude this research in Sect. 6.

2 Problem definition

2.1 Trading strategies for trading agents

Intelligent agent technology is very useful and increasingly used for developing, back-testing and evaluating automated trading techniques and program trading strategies in e-market places [16] without market costs and risks before they are deployed into the business world [10]. Currently, trading agent is mainly studied under *auction market* [18,19] and *simulated market* situations such as Supply Chain Management cases. Very few research works have been based on *continuous markets* [17] such as order-driven stock market models and *real-life market requirements*.

In fact, with the involvement of business lines in agent-based computational finance and economics studies, trading agent has potential to be customized for financial market requirements. The idea is to extend and integrate the concepts of trading agents, agent-based financial and economic computation and data mining with trading strategy development in finance to design and discover appropriate trading strategies for trading agents. Classic agent intelligence such as autonomy, adaption, collaboration and computation is also encouraged in aspects such as automated trading and trading agent collaboration for strategy integration. In this way, trading agents can dedicate to the development of financial trading strategies for market use.

A trading strategy indicates *when* a trading agent can take *what trading actions* under certain market situation. For instance, the following illustrates a general Moving Average (MA) based trading strategy.

Example 1 (MA Trading Strategy) An *MA trading strategy* is based on the calculation of moving average of security prices over a specified period of time. Let n be the length (i.e. number of prices) of MA in a time period, $P_{I,i}$ (or for short P_i) be the price of an instrument I at the time of No. i ($i < n$) price occurrence. An MA at time t (which corresponds to No. n price) is calculated as $MA_t(n)$:

$$MA_t(n) = \frac{1}{n} \sum_{i=0}^{n-1} P_i \tag{1}$$

A simple *MA strategy* is to compare the current price P_t of security I with its MA value: MA_t. Based on the conditions met, a MA strategy generates $sell$ (denoted by -1), buy (denoted by 1) or $hold$ (denoted by 0) trading signal at time t. If P_t rises above $MA_t(n)$, a buy signal is triggered, the security is then bought and held until the price falls below MA, at which time a $sell$ signal is generated and the security is sold. For any other cases, a $hold$ signal is triggered. The pseudocode of MA strategy for generating trading signal sequence S is represented as follows.

$$\begin{cases} S = 1: & \text{if } P_t > MA_t(n) \text{ and } P_i < MA_i(n) \\ & \forall\, i \in \{1, \cdots, n-1\} \\ S = -1: & \text{if } P_t < MA_t(n) \text{ and } P_i > MA_i(n) \\ & \forall\, i \in \{1, \cdots, n-1\} \\ S = 0: & \text{otherwise} \end{cases} \tag{2}$$

This strategy is usually not workable when it is employed into the real world. To satisfy the real-life needs, *market organizational factors*, *domain knowledge*, *constraints*, *trader preference* and *business expectation* are some key factors that must be involved in developing actionable trading strategies for workable trading agents. We will discuss this in the following and Sect. 3.

2.2 Searching actionable trading strategies

As the above discussed, *searching actionable trading strategies for trading agents* is a process to identify trading patterns that can reflect the 'most appropriate' combination of purchase timing, position, pricing, sizing and objects to be traded under certain market situations and interest-driving forces [15]. To this end, trading agents may cooperate with each other to either search the 'optimal' solutions from a huge amount of searchable strategy space denoted by a trading pattern. In some other cases, they collaborate to synthesize multiple trading strategy fragments favored by individual agents into an integrative strategy satisfying general concerns of each agent as well as global expectation representing trader's interest.

As we will discuss in Sect. 4.1 about data mining driven evolutionary trading agents, data mining can play a critical role in actionable strategy searching and trading pattern identification. Data mining in finance has potential in identifying not only trading signals, but also patterns indicating either iterative or repeatable occurrences. Therefore, developing actionable trading strategies for trading agents [8,12] is an interaction and collaboration process between agents and data mining [13]. The aim and objective of this process is to develop smart strategies for trading agents to take actions in the market that can satisfy trader's expectation under certain market environment.

Definition 1 (Trading Strategy) A *trading strategy* actually represents a set of individual instances Ω, which is a tuple defined as follows.

$$\begin{aligned} \Omega &= \{s_1, s_2, \ldots, s_m\} \\ &= \{(t, b, p, v, i) | t \in T, b \in B, p \in P, v \in V, i \in I)\} \end{aligned} \tag{3}$$

s_1 to s_m are instantiated trading strategies. Each of them is represented by instantiated parameters of t, b, p, v and an instrument i to be traded. $T = \{t_1, t_2, \ldots, t_m\}$ is a set of appropriate time points when trading signals are triggered; $B = \{buy, sell, hold\}$ is the set of possible behavior (i.e., trading actions) executed by trading agents. $P = \{p_1, p_2, \ldots, p_m\}$ and $V = \{v_1, v_2, \ldots, v_m\}$ are the sets of trading prices and volumes matching with corresponding trading times. $I = \{i_1, i_2, \ldots, i_m\}$ is a set of target instruments traded.

With the consideration of environment complexities and trader's favorite, the optimization of trading strategies is to search a combination set Ω' in the whole candidate set Ω, in order to achieve both user-preferred *technical* ($tech\_int()$) and *business-favored* ($biz\_int()$) *interestingness metrics* [10] in an 'optimal' or 'sub-optimal' manner. Here 'optimal' refers to the maximal/minimal (in some cases, smaller is better) values of technical and business interestingness metrics under certain market conditions and user preferences. In some situations, it is impossible or too costly to obtain 'optimal' results. For such cases, a certain level of 'sub-optimal' results are also acceptable. In this case, the sub-set Ω' indicates 'appropriate' parameter combinations of trading strategies that can support trading agents to take actions to their owner's advantage. As a result, in some sense, trading strategy optimization is to extract actionable strategies with multiple attributes towards multi-objective optimization in constrained market environment.

Definition 2 (Actionable Trading Strategy) An *actionable trading strategy* set Ω' is to achieve the following objectives:

$$\begin{aligned} tech\_int() &\rightarrow optimal\{tech\_int()\} \\ biz\_int() &\rightarrow optimal\{biz\_int()\} \end{aligned} \tag{4}$$

while satisfying the following conditions:

$$\begin{aligned} &\Omega' = \{w_1, w_2, \ldots, w_n\} \\ &\Omega' \subset \Omega \\ &m > n \end{aligned} \tag{5}$$

where $w_i (i = 1, 2, \ldots, n)$ is an instance of actionable trading strategies satisfying general $tech\_int()$ and $biz\_int()$ metrics.

The performance of actionable trading strategies should satisfy expected technical interestingness as well as business expectations under multi-attribute constraints. In the formula (4), the predicate 'optimal' is to find certain parameter combination associated with either a 'maximal' or 'minimal' optimization objective. For instance, *benefit* is maximized while *cost* is minimized. Further, the performance needs to be evaluated in terms of the background market microstructure and dynamics. Only in this way the developed trading agents can assist traders in *taking right actions at right times with right prices and volumes on right instruments*. For instance, an actionable *moving average based strategy*, say $MA(x, y)$ is a function with appropriate x and y to reach the best of expected business performance in certain market data. As a result, it generates a subset $\Omega'(MA)$ of general moving average strategies $\Omega(MA)$.

Under different situations, technical interestingness and business expectation need to be instantiated into corresponding forms. For instance, in pair mining of trading strategies [7], *coefficient* and *sharpe ratio* are used for strategy selection. In this paper, trading agent's performance is evaluated toward *enhancing benefits* while *reducing cost and risk* of host agents when they execute certain strategies in the market. This involves *strategy actionability* as discussed in the next section.

3 Enhancing trading agent actionability

Trading agent actionability, in another word, the strategy actionability taking by a trading agent, refers to the capability of a trading agent in supporting real-life trading actions that are consistent with business expectation and market constraints. In searching actionable trading strategy for trading agents, *market organizational factors* and *trader preferences* are two major issues to be concerned.

3.1 Considering market organizational factors

In practice, the development of actionable trading strategies must be based on a good understanding of *organizational factors* in a target market [21]. Otherwise it is not possible to accurately evaluate the actionable capability of the identified trading strategies as well as trading agents.

Market organization factors relevant to the trading strategy development consist of the following fundamental entities: $M = \{I, A, O, T, R, E\}$. Table 1 briefly explains these

Table 1 Market organizational factors and their relationships to actionability

Market organizational factors	Relationships to agent actionability
Traded instruments I, such as a stock or derivative, I={stock, option, feature, ...}	Varying instruments determine different trading objects and constraints on trading agents, also affect analytical methods and strategy specifications
Market participants A, A = {broker, market maker, mutual funds, ...}	Trading agents need to care for their preferences in discovering, evaluating and deploying trading strategies
Order book forms O, O = {limit, market, quote, block, stop}	Order type determines what data attributes and formats to be mined and consisted in strategies, as well as particular business interestingness for strategy evaluation
Time frame T, related to trading session, whether includes call market session or continuous session	Setting up the focusing session can greatly prune order transactions and enhance efficiency in strategy searching
Market rules R, e.g., restrictions on order execution defined by exchange and market design	They determine the validity of discovered trading strategies when deployed
Execution system E, e.g., a trading engine is order-driven or quote-driven	It limits strategy type and deployment manner after migrated to real trading system

entities and their impact on trading strategy actionability. In particular, the entity $O = \{(t, b, p, v)|t \in T, b \in B, p \in P, v \in V\}$ is further represented by attributes T, B, P and V, which are attributes of trading strategy set Ω. The elements in M form the constrained market environment of trading strategy optimization. In the strategy and system design of trading strategy optimization, we need to give proper consideration of these factors.

Any particular actionable trading strategy needs to be identified in an instantiated market niche m ($m \in M$) enclosing the above organization factors. This market niche specifies particular constraints, which are embodied through the elements in Ω and M, on trading strategy definition, representation, parameterization, searching, evaluation and deployment. The consideration of a specific market niche in trading strategy extraction can narrow down searchable strategy space in trading agent studies.

In the real world, underlying environment is more or less constrained. Constraints may be broadly embodied in terms of data, domain, interestingness and deployment aspects [9]. Constraints surrounding the development and performance evaluation of actionable trading strategy set Ω' in a particular market data set form a constraint set:

$$\Sigma = \{\delta_i^k \mid c_i \in C, 1 \leq k \leq N_i\} \tag{6}$$

where δ_i^k stands for the $k - th$ constraint attribute of a constraint type c_i, $C = \{M, D\}$ is a constraint type set covering all types of constraints in market microstructure M and data D in the searching niche, and N_i is the number of constraint attributes for a specific type c_i.

Correspondingly, actionable trading strategy set Ω' is a conditional function of Σ, which is described as:

$$\Omega' = \{(\omega, \delta)|\omega \in \Omega, \delta \in \{(\delta_i^k, a)|\delta_i^k \in \Sigma, a \in A\}\} \tag{7}$$

where ω is an 'optimal' trading pattern instance, and δ indicates specific constraints on the discovered pattern recommended to a trading agent a.

Let's explain the above concept through the illustration of *MA strategies* again. Suppose the trading strategy $MA(l)$ being deployed in the order-driven Australian Securities eXchange (ASX) market, one instantiates it into a form of $MA(5)$, which is a five-transaction MA strategy, and set a benchmark τ = AU$25.890 in trading BHP (BHP Billiton Limited) on 24 January 2007. In this situation, s/he believes $MA(5)$ is one of the most dependable $MA(l)$. Here M is instantiated into {*stock, broker, market order, continuous session, order-driven*}. In Sect. 4.1, we further discuss *evolutionary trading agents* for generating appropriate parameter combinations such as $\{l$, start date, $\tau\}$ to profitably trade a given stock in a specific constrained market.

3.2 Satisfying trader preferences

In financial markets, traders always pursue profitable and risk-free trading strategies to make good return on investment. As an assistant of a trader, a trading agent needs to care for *trader preferences* and takes actions in favor of trader's expectations. For instance, trader preferences may be embodied in terms of achieving high benefit but as low cost and risk as possible when taking certain trading positions. As the representative of traders, trading agents should target those positions with high benefit per cost.

Trading positions may vary from different market models. 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. -1 reflects either a *sell* or *holding sell* action. 0 indicates none of actions.

Further, different positions undertaken by a trading agent result in varying benefit and cost/risk. We define the concepts $benefit$ and $cost$ below to measure the performance of a trading agent when it takes certain trading strategies in a market.

Definition 3 (Benefit) *Benefit* α_s measures the cumulative payoff of a trading agent earned in undertaking position series $\{b_i\}$ in prices $\{p_i\}$ and volumes $\{v_i\}$. A position and its associated price and volume are determined by a trading strategy s.

$$\alpha_s = \sum b_i \times p_i \times v_i \tag{8}$$

Definition 4 (Cost) *Cost* β_s measures the cumulative commission and transaction costs β_i of a trading agent spent on undertaking position sequences $\{b_i\}$ determined by a trading strategy s.

$$\beta_s = \sum |b_i| \times \beta_i \times p_i \times v_i \tag{9}$$

Based on trading strategies a trading agent may follow, our goal is to *increase benefit* while *decrease cost* of a trading agent in a market scenario.

- *Increasing benefit*: increasing benefit is the positive objective of a trading agent when takes a strategy. For instance, a trading agent taking *strategy* s_1 makes higher payoff than that of taking *strategy* s_2 if $\alpha_{s_1} > \alpha_{s_2}$. Therefore, a trading agent must select trading strategies with relatively higher benefit α_s.
- *Decreasing cost*: costs reflect the negative side of a trading agent's performance in conducting a strategy. Controlling and reducing costs is one necessary step in optimizing trading strategies for trading agents. To control costs, a trading agent must select trading strategies with lower costs β_s. If $\beta_{s_1} > \beta_{s_2}$, then *strategy* s_1 is associated with higher costs.
- *Enhancing benefit per cost*: trading agents need to select those trading strategies that can guide them to take positions in the market with higher benefits while controlling costs, namely maintain the highest unit of benefit per cost $\gamma_{\alpha\beta}$.

Definition 5 (Benefit-Cost Ratio) *Benefit-cost ratio* $\gamma_{\alpha\beta,s}$ measures the unit of benefit per cost of a trading agent in undertaking position sequences $\{b_i\}$ determined by a trading strategy s.

$$\gamma_{\alpha\beta,s} = \alpha_s/\beta_s \tag{10}$$

In addition, existing financial metrics such as *sharpe ratio* [20] and empirical criteria for strategy evaluation like *beating market index return* [10,20] are also used in this research as the evaluation benchmarks of business performance.

4 Designing smart trading agents

The task 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 optimal instances from multiple classes of trading strategies into one combined powerful strategy through *collaborative trading agents*.

4.1 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 candidate strategies, and filtering optimal strategies, etc.

The strategy optimization using 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 integrating 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*

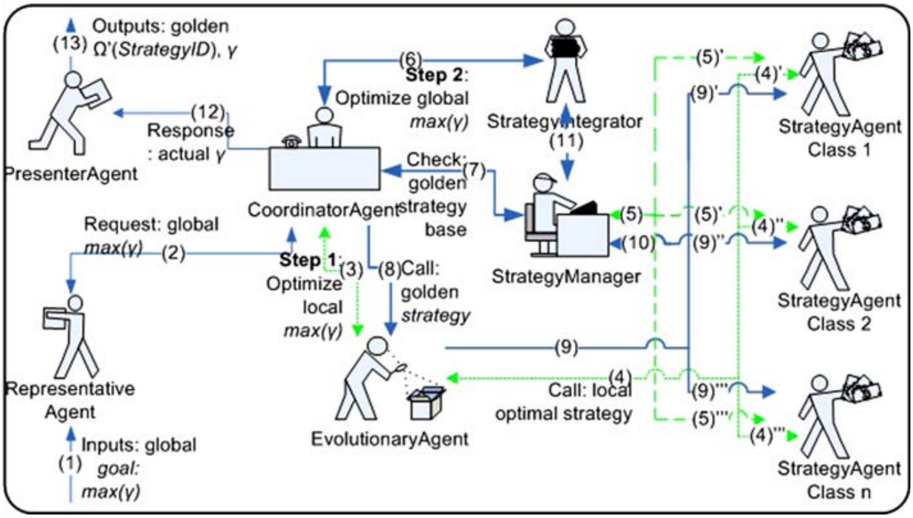


Fig. 1 Workflow of trading agent collaboration for strategy integration

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) $0 < mutrate < 1.0$

4.2 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 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. Our idea is for multiple trading agents to collaborate with each other and take concurrent positions created by multiple trading strategies to take advantage of varying strategies.

The working mechanism of *trading agent collaboration for strategy integration* is as follows. There are a few *Representive Trading Agents* in the market. Each *Representive Agent* invokes an *Evolutionary Agent* to search for an 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 objective.

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)*
)
)

Figure 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* σ. *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 the 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 an agreed negotiation model.

A simple negotiation model for the above strategy integration is as follows. In the second step, mediated by *Coordinator* agents, *StrategyIntegrator* agents discuss with *Representative* agents for the maximum number of loops for strategy optimization. Suppose *m* is the agreed loop number. For the loop j, all *StrategyAgents* are called by *Evolutionary* agents to obtain γ of the Strategy i. The *Strategy* agent (suppose its id is i) with $max(\gamma)$ is selected for this round. Its *contribution factor* σ_i is then determined. *Benefit-cost ratio* $\gamma_{\alpha\beta,s}$ for the loop i is then calculated. After executing the m loops of integration, the final *Benefit-cost ratio* is calculated. The final integrative strategy is the combination of all identified strategies in each round of m loops in terms of corresponding *contribution factors*.

$$\begin{cases} A: & \text{Identify golden strategy } i\text{:} \\ 1: & \text{let } \gamma_{\alpha\beta,s} = 0; \\ 2: & \quad \forall\, strategyID_i, i \in \{1, \cdots, n\}; \\ 3: & \quad \gamma(strategyID_i); \\ 4: & \quad i \Rightarrow \max(\gamma(strategyID_i)); \\ B: & \text{Determine contribution factor:} \\ 1: & \text{let contribution factor } \sigma_0 = 1/n; \\ 2: & \text{if } \sigma_0 >= \frac{\gamma(strategyID_i)}{\sum_{i=1}^{n}(\gamma(strategyID_i))}; \\ 3: & \quad \text{let } \sigma_i = \sigma_0 * (1 + \frac{\gamma(strategyID_i)}{\sum_{i=1}^{n}(\gamma(strategyID_i))}; \\ 4: & \text{else} \\ 5: & \quad \text{let } \sigma_i = \sigma_0 * (1 - \frac{\gamma(strategyID_i)}{\sum_{i=1}^{n}(\gamma(strategyID_i))}; \\ C: & \text{Calculate the current benefit-cost ratio:} \\ 1: & \text{for } \forall\, i \in \{1, \cdots, n\}; \\ 2: & \text{for } \forall\, j \in \{1, \cdots, m\}; \\ 3: & \text{let } \gamma_{\alpha\beta,s} = 0; \\ 4: & \quad \gamma_{\alpha\beta,s} = \gamma_{\alpha\beta,s} + \sigma_{i,j} * \gamma_{\alpha\beta,s}(strategyID_i); \\ D: & \text{Calculate the final benefit-cost ratio:} \\ 1: & \gamma_{\alpha\beta,s} = \frac{\gamma_{\alpha\beta,s}}{m}; \end{cases} \tag{11}$$

5 Real-world experiments

Since 2002, we have been working on developing trading agents and strategies with industrial partners' support, say CMCRC [1] and SIRCA [2]. Extensive experiments have been conducted on many years of multi-markets of data. An agent service-based platform F-Trade [4,11,14] has been built to support this effort. Some of our results have been delivered to partners. In this section, we illustrate some of 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 $(i = 1, 2, \ldots)$, $s \in S_i$, α_s and β_s are the $benefit$ and $cost$ of a trading agent in executing the 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* search 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:

– Trading strategies: MA, FR, CB, SR, and OBV as shown in Table 2;
– Markets and stocks: selected stocks from ASX, Hongkong, London, New York, and Japan;
– Interday trade data consisting of times, prices, volumes from 1/1/1998 to 30/12/2006;
– Training data: 2-year sliding window, say 1/1/1998–30/12/1999;
– Testing data: 1-year sliding window, say 1/1/2000–30/12/2000.

Figure 2 illustrates some results of evolutionary trading agents based optimization of 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 favorite x in a real-life market. Evolutionary trading agent is helpful for searching such a golden x. As shown in Fig. 2, the cumulative payoff with $x = 0.04$ always outperform other ds from the date 14 July 2003 in trading the listed security CBA (Australian Commonwealth Bank) in ASX market in 2003–2004.

Table 3 further shows the signals, positions, benefits and costs of trading agents following *MA-BMN Strategy*, which is an identified golden strategy by evolutionary trading agents in 2004 Hongkong Exchange data.

Table 4 shows the positions recommended by each golden strategy identified by *Collaborative Trading Agents* in 2006 Hongkong United Exchange data.

Table 2 Trading strategy base

Class	Types in a class
FR	FR-X, FR-XC, FR-XY, FR-XE
MA	MA-MN, MA-BMN, MA-CMN, MA-DMN
CB	CB-NXC, CB-NXBC
SR	SR-N, SR-NB, SR-NC, SR-NBC, SR-NDC
OBV	OBV-MN, OBV-B, OBV-C, OBV-D

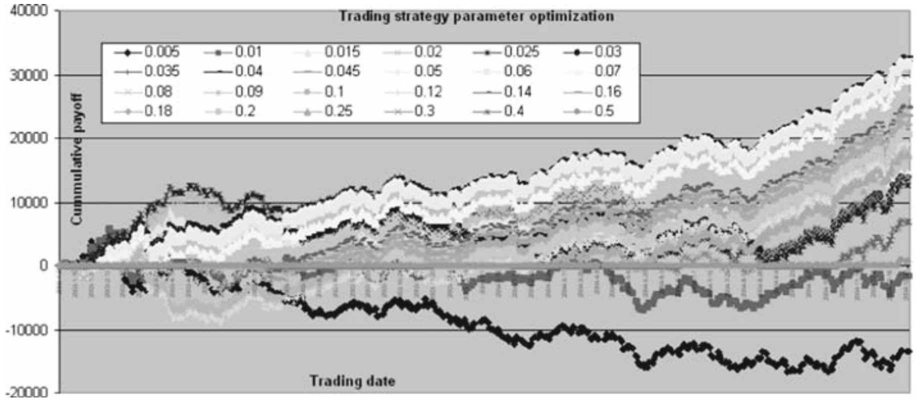


Fig. 2 Some results of evolutionary trading agent for strategy optimization

Table 3 Output excerpt of a trading strategy

Date	Price	Sell	Buy	Position	Benefit($)	Cost($)
2004-8-16	3,466	−1	0	−1	9,200	103
2004-8-17	3,480	−1	0	−1	8,850	106.5
2004-8-18	3,472	−1	0	−1	9,150	108.5
2004-8-19	3,481	−1	0	−1	8,825	110.75
2004-8-20	3,494	0	0	−1	8,500	114

Table 4 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

Figure 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 shown in Table 4, as well as the benefit (Int_Ben) of executing all golden positions concurrently recommended by *Collaborative Trading Agent* in 2003–2006 Hongkong United Exchange data.

A large amount of tests in five markets of data have shown that trading agents following our recommended golden trading strategies can obtain higher benefit-cost ratios (except FR in the first few days). In particular, collaborative trading agents concurrently executing positions recommended by individual golden strategies can greatly increase benefits while control very low costs compared with those taking positions recommended by either an individual strategy or randomly chosen strategies only (see Table 5, *lift* [12] measures how good a trading strategy is in all split data sets).

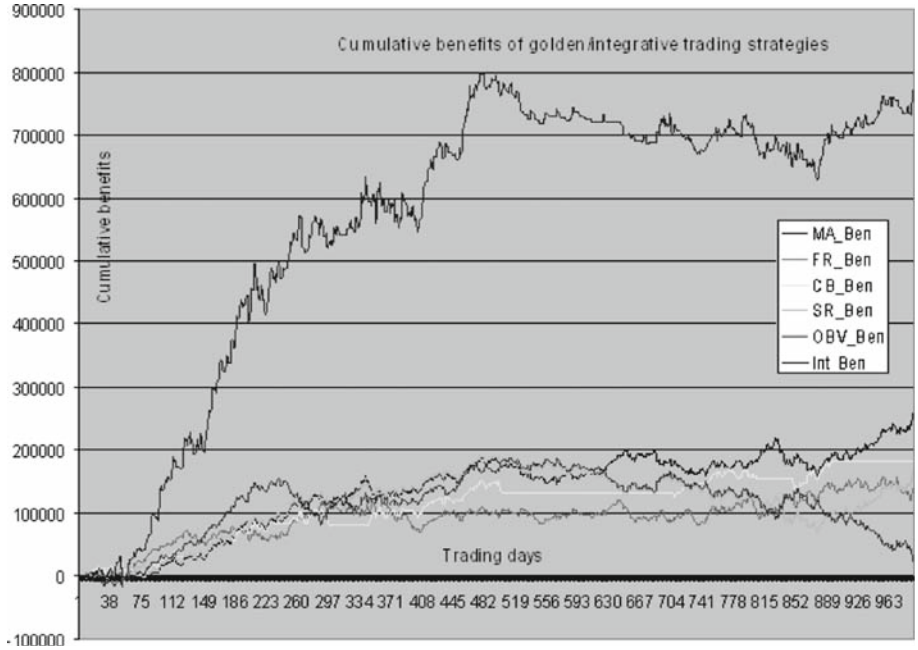


Fig. 3 Cumulative benefits of trading agents following golden trading strategies

Table 5 Lift comparison between random chosen strategies and golden strategies

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

6 Conclusions

Trading agents have demonstrated its potential in simulating market mechanism design and strategy development. However, existing trading agent research mainly focuses on artificial data and market models with limited consideration of real-life market factors and trader preferences. As a result, the studied results may not necessarily be of business interest. In fact, trading agents can contribute to traders with trading strategies that can support their smart action-taken in the market.

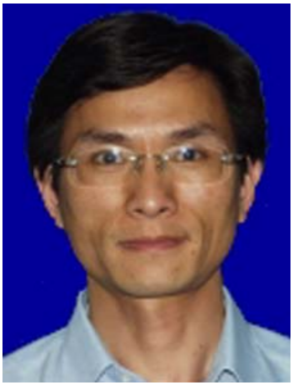
In this paper, we have demonstrated approaches to developing actionable trading strategies for trading agents. With the consideration of real-life organizational factors and trader expectations on trading agents, we have built *evolutionary trading agents* to search the golden strategies that can help an agent achieves highest *benefit-cost ratio*, and *collaborative trading agents* to concurrently trade multiple classes of identified golden trading strategies.

Extensive experiments involving hundreds of trading strategies in five markets of data have been conducted. The developed trading agents have been demonstrated to brokerage firms and business people, which have shown very promising performance from not only technical but also business perspectives. For instance, most of our golden strategies likely beat transaction costs and market index return, which are viewed as practical challenges of financial strategy development. The system supporting the above research and development, F-Trade, has been demonstrated to AAMAS2007 [11] and industrial collaborators.

References

1. Capital Markets CRC: http://www.cmcrc.com
2. SIRCA: http://www.sirca.org.au
3. Market Based Control: http://www.marketbasedcontrol.com/blog/index.php?page_id=5
4. F-Trade: http://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, Zhang C (2006) Fuzzy genetic algorithms for pairs mining. In: Proceedings on PRICAI2006, LNAI. Springer, Berlin, pp 711–720
8. Cao LB (2007) Multi-strategy integration for actionable trading agents. In: Proceedings of the International Conference on Intelligent Agent Technology. IEEE Computer Science Press, New York
9. Cao LB, Zhang CQ (2007) The evolution of KDD: towards domain-driven data mining. Int J Pattern Recognit Artif Intell 21(4):677–692
10. Cao LB, Zhang CQ (2007) Two-way significance of knowledge actionability. Int J Bus Intell Data Min
11. Cao LB, Zhang CQ (2007) 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
12. Cao LB, Luo C, Zhang CQ (2007) Developing actionable trading strategies for trading agents. In: Proceedings of the International Conference on Intelligent Agent Technology. IEEE Computer Science Press, New York
13. Cao LB, Luo C, Zhang CQ (2007) Agent-mining interaction: an emerging area. In: Proceedings of the International Conference on AIS-ADM. Springer, Berlin, pp 60–73
14. Cao LB, Zhang CQ, Zhou MC (2008) Engineering open complex agent systems: a case study. IEEE Trans. On Systems, Man, and Cybernetics–Part C: Applications and Reviews
15. Cao LB, Luo C, Luo D, Zhou MC (2008) Domain driven actionable trading evidence extraction. IEEE Trans. On Systems, Man, and Cybernetics–Part B
16. Chan T (2001) Artificial markets and intelligent agents. Ph.D. thesis, Massachusetts Institute of Technology
17. Dasgupta P, Hashimoto Y (2004) Multi-attribute dynamic pricing for online markets using intelligent agents. In: Proceedings of the International Conference on Autonomous Agents and Multiagent Systems. ACM, New York, pp 277–284
18. David E, Azoulay-Schwartz R, Kraus S (2002) Protocols and strategies for automated multi-attribute auctions. In: Proceedings of the International Conference on Autonomous Agents and Multiagent Systems. ACM, New York, pp 77–85
19. David E, Azoulay-Schwartz R, Kraus S (2003) Bidders' strategy for multi-attribute sequential English auction with deadline. In: Proceedings of the International Conference on Autonomous Agents and Multiagent Systems. ACM, New York, pp 457–464
20. Lin L, Cao LB (2008) Mining in-depth patterns in stock market. Int J Intell Syst Technol Appl
21. Madhavan A (2000) Market microstructure: a survey. J Financ Markets, 205–258
22. Wan HA, Hunter A (1997) On artificial adaptive agents models of stock markets. Simulation 68:279–289
23. Wellman MP et al (2001) Designing the market game for a trading agent competition. IEEE Internet Comput 5(2):43–51

Author Biographies



Longbing Cao is an Associate Professor of Information Technology with the University of Technology Sydney, and the Data Mining Research Leader of Australian Capital Markets Cooperative Research Centre. His research interests include data mining, multi-agent technology, and multiagent and data mining integration.

Tony He is an Associate Professor of Finance with the University of Technology, Sydney. His research interests include financial market modelling, nonlinear economic dynamics, behavior finance and asset pricing.