

A novel method for coevolving PS-optimizing negotiation strategies using improved diversity controlling EDAs

Jeonghwan Gwak · Kwang Mong Sim

Published online: 21 September 2012
© Springer Science+Business Media, LLC 2012

Abstract In agent-mediated negotiation systems, the majority of the research focused on finding negotiation strategies for optimizing price only. However, in negotiation systems with time constraints (e.g., resource negotiations for Grid and Cloud computing), it is crucial to optimize either or both price and negotiation speed based on preferences of participants for improving efficiency and increasing utilization. To this end, this work presents the design and implementation of negotiation agents that can optimize both price and negotiation speed (for the given preference settings of these parameters) under a negotiation setting of complete information. Then, to support negotiations with incomplete information, this work deals with the problem of finding effective negotiation strategies of agents by using coevolutionary learning, which results in optimal negotiation outcomes. In the coevolutionary learning method used here, two types of estimation of distribution algorithms (EDAs) such as conventional EDAs (S-EDAs) and novel *improved dynamic diversity controlling* EDAs (ID²C-EDAs) were adopted for comparative studies. A series of experiments were conducted to evaluate the performance for coevolving effective negotiation strategies using the EDAs. In the experiments, each agent adopts three representative preference criteria: (1) placing more emphasis on optimizing more price, (2) placing equal emphasis on optimizing ex-

act price and speed and (3) placing more emphasis on optimizing more speed. Experimental results demonstrate the effectiveness of the coevolutionary learning adopting ID²C-EDAs because it generally coevolved effective converged negotiation strategies (close to the optimum) while the coevolutionary learning adopting S-EDAs often failed to coevolve such strategies within a reasonable number of generations.

Keywords Negotiation agents · Price and negotiation speed optimization · Coevolutionary learning · Estimation of distribution algorithms · Dynamic diversity control

1 Introduction

In distributed systems involving the interactions of autonomous agents on behalf of their owners, negotiation activities are essential for resolving differences and conflicting goals [11] and to control and manage resources [30, 34, 38]. Automated negotiation among agents has been widely used for supporting e-commerce and is also becoming increasingly important for managing massive distributed computational systems such as Grid/Cloud computing systems because interactions between participating agents can occur in many different contexts. Whereas there are many existing negotiation agents for e-commerce (e.g., [4, 22]), Grid resource management (e.g., [2, 20, 30, 31, 34, 38]) and Cloud resource management (e.g., [33, 35, 36]), (1) most of the negotiation agents are designed to reach an agreement consisting of coinciding proposals of participating agents and (2) each agent's decision to reach an agreement is focused on optimizing the value of the proposal (typically price) only without consideration of reaching a consensus more rapidly

J. Gwak
School of Information and Mechatronics, Gwangju Institute of Science and Technology, 261 Cheomdan-gwagiro (Oryong-dong), Buk-gu, Gwangju 500-712, Republic of Korea
e-mail: gawk@gist.ac.kr

K.M. Sim (✉)
School of Computing, The University of Kent, Chatham Maritime, Chatham, Kent ME4 4AG, UK
e-mail: prof_sim_2002@yahoo.com

(i.e., the participating agents do not consider optimizing negotiation speed). However, there are some practical negotiation applications with time constraints in which both the issues of obtaining the cheapest possible resources and getting them rapidly are essential (e.g., negotiations for Grid or Cloud resources). In such applications, obtaining resources more rapidly is one of the most desirable properties depending on negotiation participants preferences because any delay incurred on waiting for negotiations as well as resources can be perceived as an overhead. Even though there is a lot of existing research (e.g. [14, 16, 27]) that focuses on developing multi-attribute negotiation mechanisms to deal with different attributes (i.e., issues of negotiation such as price, quality, quantity, delivery time, etc.) of participating negotiation agents, there is little research that considers the duration of a negotiation (i.e., negotiation speed) as a factor affecting performance for time-constrained negotiations [32]. Whereas (negotiation) success rate (i.e., the chance of successfully finding a mutually acceptable agreement) is the main consideration for negotiation agents that operate in domains that do not have very stringent constraints on time, negotiation speed (as well as success rate) is an important consideration for those that operate in domains with very stringent constraints on time. This is because negotiation agents that consider enhancing negotiation speed can make agreements quickly (by sacrificing expected utility on issues of negotiation) and therefore, it is also possible for the agents to obtain higher success rates in negotiations under such time-constrained domains. In this regard, designing negotiation agents considering negotiation speed and finding efficient (or optimal) negotiation strategies of such agents are the main focuses of this work. Even though this work currently deals with negotiation considering negotiation speed based on a single issue (i.e., price), it can be extended to deal with multi-attribute negotiation considering negotiation speed.

In this work, the agents focusing on *optimizing price only* and *optimizing both price and negotiation speed* are denoted as *price optimizing* (P-optimizing) and *price and speed optimizing* (PS-optimizing) ones, respectively. PS-optimizing agents were first proposed and considered in [32]. To illustrate the detailed negotiation applications with examples, consider the following negotiation scenarios in which: (1) there are two types of self-interested PS-optimizing negotiation agents (that they will act so as to maximize their own outcomes) called as a seller (or consumer) and buyer (or provider) and (2) each seller and buyer has different preference criteria for optimizing both price and negotiation speed. The preference criteria of the seller can be classified into the following two cases.

- (1) The seller prefers to sell (or provide) a resource/service at a higher price than the given (expected) agreement price at the expense of having to wait longer than the

given (expected) agreement time. We denote such seller is more P-optimizing (*more-P-optimizing*).

- (2) The seller prefers to sell (or provide) a resource/service more rapidly than the given (expected) agreement time perhaps by providing its resource/service with a lower price than the given (expected) agreement price at an earlier negotiation round. We denote such seller is more speed optimizing (*more-S-optimizing*).

Similarly, the preference criteria of the buyer can also be classified into the following two cases.

- (1) The buyer prefers to acquire cheaper resource/service alternatives than the given (expected) agreement price at the expense of having to wait longer than the given (expected) agreement time. We denote the buyer is *more-P-optimizing*.
- (2) The buyer prefers to acquire a resource/service more rapidly than the given (expected) agreement time perhaps by paying a higher price than the given (expected) agreement price at an earlier negotiation round. We denote the buyer is *more-S-optimizing*.

To adequately address such negotiation problems, negotiation agents called PS-optimizing agents should be designed to: (1) determine the solution space SS_{PS-opt} for PS-optimizing negotiation consisting of (i) the solution space SS_{NP} for optimizing price (in which different possible preference criteria of price can be represented) and (ii) the solution space SS_{NS} for optimizing negotiation speed (in which different possible preference criteria of negotiation speed can be represented), (2) appropriately optimize both price and negotiation speed in SS_{PS-opt} for the given various combinations of possible preference criteria (of agents), and (3) make successful agreements in various negotiation situations. To this end, the impetus of this work is to devise mechanisms for finding effective PS-optimizing negotiation strategies (of agents) which result in reasonable PS-optimizing negotiation outcomes.

Based on the information about their opponents (i.e., the other participating agents), negotiation (parameter) settings can be generally classified into the two types: (1) a *complete information setting* in which (participating) agents share their private information to their opponents, and (2) an *incomplete information setting* in which agents do not share their private information to their opponents. We denote the agent having the negotiation setting of a complete information setting (respectively, an incomplete information setting) as the *agent with complete information* (respectively, the *agent with incomplete information*). Following the above definitions, PS-optimizing agents are also divided into two categories based on the negotiation settings that they adopt. That is, a PS-optimizing agent under a complete information setting knows its opponent's private information while

a PS-optimizing agent under an incomplete information setting does not know its opponent's private information. Further details of negotiation models for P-optimizing negotiation and the PS-optimizing negotiation will be described and compared in Sect. 2.

The existing preliminary works in [32] and [9] have attempted to find negotiation strategies for PS-optimizing agents with incomplete information using coevolutionary learning by adopting evolutionary algorithms (EAs). Nevertheless, the results in [32] and [9] showed that: (1) there are possibilities of coevolution failure using the fitness function defined in [32] due to the ambiguity in the utility space, (2) the converged coevolution results cannot be achieved in some cases using conventional EAs used in [32] and [9]. Furthermore, in [32] and [9], there was no theory explaining and supporting the optimality of the achieved results. To overcoming these drawbacks and to complement and enhance the existing PS-optimizing agents, this work will design: (1) PS-optimizing agents for performing effective PS-optimizing negotiations under a complete information setting (Sect. 4.1) and (2) mechanisms for finding effective negotiation strategies of PS-optimizing agents with incomplete information (Sect. 4.2) by using coevolutionary learning (described in Sect. 3) adopting estimation of distribution algorithms (EDAs).

A series of experiments (see Sect. 5) was carried out to: (1) show the effectiveness of the coevolutionary learning for finding effective negotiation strategies of PS-optimizing agents with incomplete information and (2) compare the performance of coevolutionary learning adopting S-EDAs against adopting ID²C-EDAs. Empirical results in Sect. 5 show that ID²C-EDAs can coevolve effective converged negotiation strategies which are close to the optimum for both PS-optimizing agents in most of the cases. While Sect. 6 compares this work with existing works, Sect. 7 concludes this paper by summarizing a list of contributions and future work.

2 Negotiation models

This work considers a bilateral negotiation model between two self-interested agents with conflicting interests such that the seller (S) that wishes to provide a good or service at the highest possible price and the buyer (B) that purchase the good or service at the cheapest possible price. We first investigate one of the most widely used P-optimizing negotiation model for optimizing price only. Then, the P-optimizing negotiation model will be extended to the PS-optimizing negotiation model that is capable of dealing with optimizing both price and negotiation speed using preferences of price and negotiation speed.

2.1 Price optimizing negotiation model

In the P-optimizing negotiation model, there are three key elements of negotiation [15]: (1) the negotiation protocol, (2) the negotiation strategies that the agents adopt during the negotiation process, and (3) the utility functions for the agents. The agents adopt Rubinstein's alternating offers protocol [26] and negotiate by exchanging proposals with their negotiation partners. The alternating offers protocol is simple but it is the most influential general negotiation protocol. Furthermore, it has been applied to many existing works (e.g., see [6, 21, 41]). At each alternate round, an agent makes and sends a proposal. Then, the other agent evaluates the proposal and takes one of the following actions: (1) accepting the proposal, (2) rejecting the proposal, or (3) making a counter proposal. Negotiation between the two agents terminates with an agreement when an offer or a counter-offer is accepted or with a conflict if no agreement is reached when one of the two agents' deadlines is reached. An agreement is reached when one agent proposes a deal that matches or exceeds what another agent asks for.

The agent $x \in \{B, S\}$ generates a proposal at a negotiation round t , $0 \leq t \leq \tau_x$, as follows:

$$P_t^x = IP_x + (-1)^\alpha \left(\frac{t}{\tau_x} \right)^{\lambda_x} |RP_x - IP_x|, \quad (1)$$

where $\alpha = 1$ for S and $\alpha = 0$ for B . IP_x is the initial price of x that is the most favorable price for x , and RP_x is the reserve price that is the least favorable price for x ; τ_x is the deadline and λ_x , $0 \leq \lambda_x \leq \infty$, is the time-dependent strategy of x . During the negotiation process, starting from the initial prices, successive proposals of S are monotonically decreasing while successive proposals of B are monotonically increasing.

As shown in Fig. 1, for each agent x , the possible range of price, $[IP_x, RP_x]$, is denoted as the acceptability zone for price of x , $AccZ_x^{NP}$, and the possible range of negotiation time, $[0, \tau_x]$, is denoted as the acceptability zone for negotiation time of x , $AccZ_x^{NT}$. The negotiation solution space (NSS) for the negotiation between B and S consists of: (1) the agreement zone of price (AgZ^{NP}), or sometimes called the price-surplus, which is the overlapping region between $AccZ_B^{NP}$ and $AccZ_S^{NP}$, and (2) the agreement zone of negotiation time (AgZ^{NT}) which is the overlapping region between $AccZ_B^{NT}$ and $AccZ_S^{NT}$. In Fig. 1, AgZ^{NP} is $[RP_S, RP_B]$ and AgZ^{NT} is $[0, \min\{\tau_B, \tau_S\}]$.

Time-dependent negotiation strategies are adopted in which the negotiation agents make successive proposals depending on the remaining negotiation time. The concession behavior of x is determined by the values of the time-dependent strategy and is classified as follows [28, 29, 37]:

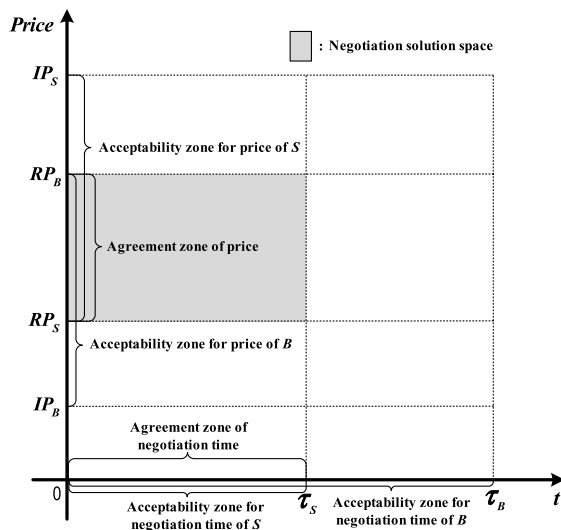


Fig. 1 Example of negotiation solution space between B and S

- (1) Conciliatory ($0 < \lambda_x < 1$): x makes larger concessions in earlier negotiation rounds and smaller concessions in later negotiation rounds.
- (2) Linear ($\lambda_x = 1$): x makes a constant rate of concession.
- (3) Conservative ($1 < \lambda_x < \infty$): x makes smaller concessions in earlier negotiation rounds and larger concessions in later negotiation rounds.

Let D be the event in which x fails to reach an agreement. The utility function of x is defined as $U_x : [IP_x, RP_x] \cup D \rightarrow [0, 1]$ such that $U_x(D) = 0$ and for any $P_t^x \in [IP_x, RP_x]$, $U_x(P_t^x) > U_x(D)$ in which $U_x(P_t^x)$ is given as follows:

$$U_x(P_t^x) = u_{\min} + (1 - u_{\min}) \left(\frac{RP_x - P_t^x}{RP_x - IP_x} \right), \quad (2)$$

where u_{\min} is the minimum utility that x receives for reaching an agreement at RP_x and the value of u_{\min} is set larger than 0. u_{\min} is set to 0.001 in this work for the experimental purpose. Then, at $P_t^x = RP_x$, $U_x(RP_x) = 0.001 > U_x(D) = 0$.

Definition 1 (P-optimizing Agent) For a given negotiation setting, a P-optimizing agent is designed to optimize the price only by maximizing the utility in (2).

A negotiation between P-optimizing agents is denoted as the P-optimizing negotiation. Self-interested P-optimizing agents B and S favor an agreement that maximizes their own (price) utilities given in (2) at an agreement price.

In P-optimizing negotiations between B and S , finding their optimal negotiation strategies plays an important role in a sense that by adopting optimal negotiation strategies, both achieve optimal negotiation outcomes (i.e., optimal agreement prices). In determining optimal negotiation

strategies for P-optimizing negotiations with complete information settings, deadline effect is the most important factor. This is because if one P-optimizing agent has a longer deadline than the other, the agent having a longer deadline will dominate the whole negotiation. Since the strategy of the agent having a longer deadline determines whether both agents can reach an agreement before their deadlines, the agent having a longer deadline has (significant) a bargaining advantage in terms of time over the other agent.

For a P-optimizing negotiation under a complete information setting, an agent knows the other agent’s private information such as RP and deadline. Therefore, the optimal agreement price (P_c^{P-opt}) and agreement time (T_c^{P-opt}) for the P-optimizing negotiation between B and S can be analyzed by the following theorems.

Theorem 1 [40, pp. 199–200] *If the P-optimizing agent B has longer deadline than the P-optimizing agent S , P_c^{P-opt} is RP_S and T_c^{P-opt} is τ_S .*

Proof Since the minimal possible agreement price for B is RP_S and at which B obtains the maximal utility, P_c^{P-opt} is made at RP_S . Whatever strategy S adopts, S concedes to RP_S at τ_S following (1). Before reaching τ_S , the utility of S ’s proposals for B will be lower than the utility at τ_S . Furthermore, B fails to reach an agreement after τ_S . Hence, T_c^{P-opt} is made at τ_S . \square

Theorem 2 [40, pp. 199–200] *If the P-optimizing agent S has longer deadline than the P-optimizing agent B , P_c^{P-opt} is RP_B and T_c^{P-opt} is τ_B .*

Proof Symmetrically, P_c^{P-opt} is made at RP_B and T_c^{P-opt} is made at τ_B because at which S obtains its maximal utility. \square

Finally, using the obtained P_c^{P-opt} and T_c^{P-opt} from Theorems 1 and 2, the optimal P-optimizing negotiation strategy of $B(\lambda_B^{P-opt})$ and the optimal P-optimizing negotiation strategy of $S(\lambda_S^{P-opt})$ are derived from (1) by substituting P_t^x in (1) by P_c^{P-opt} and t in (1) by T_c^{P-opt} , respectively, as follows:

$$\lambda_B^{P-opt} = \log_{\left(\frac{T_c^{P-opt}}{\tau_B}\right)} \left(\frac{P_c^{P-opt} - IP_B}{RP_B - IP_B} \right), \quad (3)$$

$$\lambda_S^{P-opt} = \log_{\left(\frac{T_c^{P-opt}}{\tau_S}\right)} \left(\frac{IP_S - P_c^{P-opt}}{IP_S - RP_S} \right). \quad (4)$$

Theorems 1 and 2 are based on Theorems 1 and 2 in [40, pp. 199–200].

2.2 Price and speed optimizing negotiation model

The proposed PS-optimizing negotiation model also considers price as a negotiation issue similarly to the P-optimizing negotiation model in Sect. 2.1. Therefore, the agents participating in PS-optimizing negotiations exchange offers or counter-offers that consist of price proposals only—but not time proposals or both price and time proposals—during the negotiation process. However, compared to the P-optimizing negotiation model designed to take only price into consideration in optimizing negotiation outcomes, the PS-optimizing negotiation model is designed to take both price and negotiation speed (in terms of the number of negotiation rounds) into consideration in optimizing negotiation outcomes.

Definition 2 (PS-optimizing Agent) For a given negotiation setting, a PS-optimizing agent is designed to optimize both price and negotiation speed (by maximizing the total utility consisting of both price and speed utilities) using its given preferences of price and negotiation speed.

A negotiation between PS-optimizing agents is denoted as the PS-optimizing negotiation.

The PS-optimizing negotiation model also has three key elements similarly to the P-optimizing negotiation model. The PS-optimizing agents also adopt Rubinstein’s alternating offers protocol and time-dependent negotiation strategies as in P-optimizing agents. However, for each PS-optimizing agent, there are two types of utility functions: (1) one designed to measure the degree of satisfaction for price and (2) the other designed to measure the degree of satisfaction for negotiation speed. Depending on the strategies that a PS-optimizing agent adopts, there can be a variety of possible PS-optimizing negotiation outcomes. Hence, this research will focus on designing PS-optimizing agents *B* and *S* and finding their optimal PS-optimizing negotiation strategies for achieving optimal PS-optimizing negotiation outcomes for their given preferences of price and negotiation speed.

In addition to the three key elements, the PS-optimizing negotiation model has one more key element: the preferences of price and negotiation speed (for each PS-optimizing agent). With regard to preferences of price and negotiation speed of a PS-optimizing agent, different preference criteria such as optimizing price and optimizing negotiation speed are individually modeled as corresponding different weightings of price and negotiation time, respectively; for a PS-optimizing agent *x*, the preference for optimizing price is denoted as w_{NP}^x and the preference for optimizing negotiation speed is denoted as w_{NS}^x . Based on the user’s preferences for optimizing price and optimizing speed, w_{NP}^x and w_{NS}^x are provided by the user with the constraint $w_{NP}^x + w_{NS}^x = 1.0$ where $w_{NP}^x \geq 0$ and $w_{NS}^x \geq 0$. $w_{NP}^x + w_{NS}^x$ is set

to 1.0 because the preference criteria of *x* are interdependent and conflicting each other; as *x* prefers to achieve negotiation outcomes that is more P-optimizing at the expense of waiting longer, then *x* will put more emphasis on w_{NP}^x and less emphasis on w_{NS}^x . Conversely, as *x* prefers to achieve its negotiation outcome more rapidly at the expense of conceding more in price, then *x* will put more emphasis on w_{NS}^x and less emphasis on w_{NP}^x . Depending on different preference criteria, agents can be summarized as the following three representative groups [32]:

- (1) (*Totally*) P-optimizing agents in which total emphasis is given for optimizing price such as $(w_{NP}^x, w_{NS}^x) = (1.0, 0.0)$.
- (2) (*Totally*) S-optimizing agents in which total emphasis is given for optimizing negotiation speed such as $(w_{NP}^x, w_{NS}^x) = (0.0, 1.0)$.
- (3) PS-optimizing agents in which emphases are given for optimizing both price and negotiation speed such as $(w_{NP}^x, w_{NS}^x) = \{\text{the weightings except } (1.0, 0.0) \text{ and } (0.0, 1.0)\}$.

However, (*Totally*) S-optimizing agents are not considered because such S-optimizing agents model the situation when agents are totally optimizing negotiation speed without consideration of optimizing price. This negotiation situation will not be realistic in practice because such S-optimizing agents generally reach an agreement without any negotiation by just accepting its opponent’s first proposal. Hence, the possible region of preferences of price and negotiation speed is set as $(w_{NP}^x, w_{NS}^x) = [(1.0, 0.0), (0.0, 1.0)]$.

In regard to various possible combinations of preference criteria of PS-optimizing agents, there are three representative groups of PS-optimizing agents: (1) the agents placing the equal emphasis on optimizing price and negotiation speed such as $(w_{NP}^x, w_{NS}^x) = (0.5, 0.5)$ are denoted as *exact*-PS-optimizing agents, (2) if agents place more emphasis on optimizing price than *exact*-PS-optimizing agents, then they are denoted as *more*-P-optimizing agents, and (3) if agents place more emphasis on optimizing negotiation speed than *exact*-PS-optimizing agents, then they are denoted as *more*-S-optimizing agents.

Comparing to P-optimizing agents, the PS-optimizing agent *x* requires two types of utility functions: (1) a price utility function for measuring the degree of satisfaction in terms of price and (2) a speed utility for measuring the degree of satisfaction in terms of negotiation speed. The price utility function U_{NP}^x for the given input P_x (for price) and the speed utility function U_{NS}^x for the given input T_x (for negotiation time) are defined as follows:

$$U_{NP}^x(P_x) = \begin{cases} u_{\min}^P + (1 - u_{\min}^P) \left(\frac{RP_x - P_c}{RP_x - IP_x} \right), & \text{if an agreement is reached at } P_c, \\ 0, & \text{otherwise,} \end{cases} \quad (5)$$

$$U_{NS}^x(T_x) = \begin{cases} u_{\min}^S + (1 - u_{\min}^S)(1 - \frac{T_c}{T_x}), & \text{if an agreement is reached at } T_c, \\ 0, & \text{otherwise,} \end{cases} \quad (6)$$

where $U_{NP}^x(P_x) \in [0, 1]$ and $U_{NS}^x(T_x) \in [0, 1]$. u_{\min}^P is the minimum utility that x receives a deal at its RP , and u_{\min}^S is the minimum utility that x receives a deal at its deadline. For the experimental purpose, the values of u_{\min}^P and u_{\min}^S is set to 0.0001. Next, for achieving the composite utility of x consisting of both U_{NP}^x and U_{NS}^x , the following total utility function U_{Total}^x was used:

$$U_{Total}^x(P_x, T_x) = w_{NP}^x \times U_{NP}^x(P_x) + w_{NS}^x \times U_{NS}^x(T_x) \quad (7)$$

where $P_x \in \{0, P_c\}$ and $T_x \in \{0, T_c\}$. If x does not reach an agreement before its deadline, then $U_{Total}^x = 0$ because $U_{NP}^x = U_{NS}^x = 0$. If x reaches an agreement at $P_x = P_c$ and $T_x = T_c$, then $U_{Total}^x(P_c, T_c) > 0$ because $U_{NP}^x > 0$ and $U_{NS}^x > 0$.

The remaining part is designing PS-optimizing agents and finding their optimal negotiation strategies to achieve optimal PS-optimizing negotiation outcomes under both complete and incomplete information settings. In designing PS-optimizing agents, the design goal is as follows:

Design goal The ultimate design goal of PS-optimizing agents is to achieve optimal PS-optimizing negotiation outcomes satisfying preferences of price and negotiation speed under given negotiation settings. Even though a PS-optimizing negotiation cannot achieve optimal negotiation outcomes, the performance of the PS-optimizing negotiation in optimizing the preferences should be *superior to* or (at least) *equal to* that of the P-optimizing negotiation. The latter case is denoted as the *minimum performance requirement* in this work.

The similarities and differences between P-optimizing agents (Sect. 2.1) and PS-optimizing agents (Sect. 2.2) are as follows. (1) Both negotiation models adopt Rubinstein’s alternating offers protocol as the negotiation protocol. Furthermore, agents in the two negotiation models exchange offers or counter-offers that consist of price proposals—but not time proposals or both price and time proposals—for making a mutual agreement. (2) To evaluate price and negotiation time (of the proposals), a speed utility function as well as a price utility function is required. Accordingly, the total utility function consisting of both price and speed utility functions, associated with preferences of price and negotiation speed, respectively, is adopted for PS-optimizing agents. However, for P-optimizing agents, the sole utility function equivalent to the price utility function in (2) is used. (3) As the name denotes, the PS-optimizing negotiation model requires an optimization procedure (denoted as the PS-optimization) for optimizing both price and negotiation speed while the (original) P-optimizing negotiation

model in itself have the optimization procedure (denoted as the P-optimization) for optimizing price only. Hence, a PS-optimizing negotiation mechanism that enables rational PS-optimization is required to find effective negotiation strategies of PS-optimizing agents.

In designing the PS-optimizing negotiation mechanism, it is assumed that PS-optimizing agents do not change their preferences of price and negotiation speed with the knowledge of the opponent’s information. This means that the PS-optimizing agents with their given preferences of price and negotiation speed are cooperative for optimizing both price and negotiation speed. This assumption makes sense in that PS-optimizing negotiations will not operate well if negotiating agents do not cooperate at all. For instance, if one agent having a bargaining advantage in terms of time tries to achieve higher speed utility without conceding any price utility, there is no reason for its opponent to make an earlier agreement by conceding its speed utility. Furthermore, PS-optimizing agents should be trustable in the sense that they cooperate for optimizing negotiation speed without changing its initial preferences of price and negotiation speed during negotiation process. Another issue for designing PS-optimizing agents is to determine: (1) the value(s) of preferable agreement price in SS_{NP} using w_{NP}^x and (2) the value(s) of preferable agreement time in SS_{NS} using w_{NS}^x . Determining such a range of values within effective SS_{PS-opt} (consisting of SS_{NP} and SS_{NS}) is essential because there can be different realizations of PS-optimizing negotiations depending on the values. The specific details for designing PS-optimizing agents to find optimal negotiation strategies will be described in Sect. 4.

3 Overview of EDAs for coevolutionary learning

EDAs, sometimes called probabilistic model building genetic algorithms (PMBGAs), have become one of the new paradigms within genetic and evolutionary computation research [17, 42]. Like other EAs based on ideas borrowed from genetics and natural selection such as genetic algorithms (GAs), evolutionary strategies (ESs) and evolutionary programming (EP), EDAs also use selection to choose good candidate solutions and successively evolve a population of the selected solutions until some termination criteria are satisfied. However, to evolve a population of promising solutions, EDAs build probabilistic models of the selected solutions and sample useful genetic information (i.e., good offspring) from the probabilistic models instead of using variation operators such as crossover and mutation. From the perspective of the fitness landscape (that is the geographical distribution consisting of peaks and valleys of fitness over solution space), while EAs such as GAs, ESs and EP search promising regions (i.e., solutions) of fitness landscape with

Table 1 Symbols used for the EDAs

g	The number of generations
G^{\max}	The (predefined) maximum number of generations
cnt	The number of evolution restarts
CNT^{\max}	The (predefined) maximum number of evolution restarts
P_g	The population at generation g
S_g	The subset of P_g selected using the selection operator
$f_{\mathbf{x}^g}(\mathbf{x}^g)$	The probability distribution estimated from S_g
O_g	The offspring generated by sampling $f_{\mathbf{x}^g}(\mathbf{x}^g)$
n_P	The number of individuals consisting of P_g (i.e., the size of P_g)
n_S	The number of individuals consisting of S_g (i.e., the size of S_g)
n_O	The number of individuals consisting of O_g (i.e., the size of O_g)
f_{best}^g	The fitness value of the best individual in P_g
f_{avg}^g	The average fitness value of individuals in P_g
$\text{Var}(P_g)$	The variance of solution values of individuals in P_g
$\text{Div}(P_g)$	The population diversity of P_g
n_{band}	The number of bands for the (entire) solution space
$Band_Size$	The size of a (fixed) band
FB_List	The list of feasible bands
FI_List	The list of feasible individuals
$G^{\max_infeasible_band}$	The maximum number of generations required to determine infeasible bands
α	The (minimum) threshold of fitness difference (used for similarity measure)
β	The (minimum) threshold of solution difference (used for similarity measure)
δ_{low}	The threshold for the lowest possible $\text{Div}(P_g)$ (used for controlling diversity)
δ_{high}	The threshold for the highest possible $\text{Div}(P_g)$ (used for controlling diversity)
δ_{fit}	The (minimum) threshold of the absolute fitness difference between f_{best}^g and f_{avg}^g (used in determining termination of EDAs)
δ_{var}	The (minimum) threshold of $\text{Var}(P_g)$ (used in determining termination of EDAs)

both exploitation and exploration using genetic operators (i.e., selection and variation operators), EDAs search the regions by exploiting feasible probabilistic models and efficiently traversing the solution space [1, 25].

This section demonstrates the application of EDAs to solve the coevolutionary problem of finding optimal negotiation strategies of PS-optimizing agents operating under an incomplete information setting. First, S-EDA is presented. Then, ID²C-EDA incorporating S-EDA with a novel diversity controlling technique is presented. Table 1 shows symbols used for the EDAs in this work.

3.1 Description of S-EDA

The S-EDA is based on the continuous (i.e., real-coded) univariate marginal distribution algorithm (UMDA_c) [17]. The pseudocode of S-EDA is presented as follows.

Step 1. Initialization

Generate the initial population P_0 with n_P individuals at random;
 $g \leftarrow 0$; $cnt \leftarrow 0$.

Step 2. Selection

$g++$;

Select a set of promising candidates S_{g-1} with $n_S (< n_P)$ individuals from P_{g-1} .

Step 3. Building Model

Estimate the probability distribution $f_{\mathbf{x}^g}(\mathbf{x}^g)$ from S_{g-1} .

Step 4. Sampling Model

Generate offspring O_g with n_O individuals by sampling $f_{\mathbf{x}^g}(\mathbf{x}^g)$.

Step 5. Replacement

Create a new population P_g by replacing some individuals of P_{g-1} with O_g .

Step 6. Reinitializing Population and Restarting Evolution

If $cnt < CNT^{\max}$ and an inappropriate configuration is detected,

initialize P_g at random;

$g \leftarrow 0$; $cnt++$;

Go to Step 2.

Step 7. Termination

If the termination criteria are not satisfied,
 go to Step 2.

Else return the best solution found so far.

The main distinguishing features of S-EDA (compared to other EAs adopting genetic operators) is building a probabilistic model (Step 3) and sampling the model to generate new solutions, i.e., offspring (Step 4). A continuous optimization problem with n variables is considered. The corresponding n -dimensional random variable and one of its possible instances at each generation g is denoted as $\mathbf{X}^g = (X_1^g, X_2^g, \dots, X_n^g)$ and $\mathbf{x}^g = (x_1^g, x_2^g, \dots, x_n^g)$. Following UMDA_c, S-EDA assumes that marginal independence among the variables. Hence, the joint probability distribution of \mathbf{X}^g follows an n -dimensional normal distribution which is factorized as a product of n independent univariate marginal distributions as follows.

$$f_{\mathbf{X}^g}(\mathbf{x}^g) = \prod_{i=1}^n f_{X_i^g}(x_i^g).$$

Each variable of \mathbf{X}^g follows a univariate normal distribution with mean μ_i^g and the standard deviation σ_i^g as follows:

$$f_{X_i^g}(x_i^g) = \frac{1}{\sqrt{2\pi}\sigma_i^g} e^{-\frac{(x_i^g - \mu_i^g)^2}{2(\sigma_i^g)^2}}, \quad \text{with } i = 1, 2, \dots, n.$$

μ_i^g and σ_i^g are estimated using maximum likelihood estimation from S_{g-1} as follows:

$$\mu_i^g = \frac{1}{n_S} \sum_{j=1}^{n_S} (x_i^{g-1})_j \quad \text{and}$$

$$\sigma_i^g = \sqrt{\frac{1}{n_S} \sum_{j=1}^{n_S} ((x_i^{g-1})_j - \mu_i^g)^2},$$

where $(x_i^{g-1})_j$ is the j -th individual in S_{g-1} .

Then, offspring O_g are randomly generated by sampling normal random variables from $f_{\mathbf{X}^g}(\mathbf{x}^g)$.

Another distinguishing feature of S-EDA (compared to UMDA_c, as well as other conventional EAs) is reinitializing population and restarting the evolution (Step 6). This is to escape an inappropriate configuration of populations (where S-EDA can no longer evolve a population containing promising solutions for future evolution process) though simply restarting evolution process with randomly initialized population. In the coevolutionary learning problem (described in Sect. 4.2 in detail), there can be two types of inappropriate population configurations: (1) *Type-I error*: P_g cannot converge to a certain value until G^{\max} is reached (i.e., very slow convergence or non-convergence is presented) and (2) *Type-II error*: This is due to premature convergence occurring at early generations (generally,

$g \leq G^{\max_infeasible_band}$) caused by the domination of inappropriate individuals in P_g having fitness values of all 0s or all 1s (that cannot occur in the coevolution problem described in Sect. 4.2), which occurs from repeated inappropriate random pairing of individuals [25]. If an inappropriate populations is detected, S-EDA initializes P_g and restarts its evolution procedures. Step 6 is incorporated before testing termination criteria in Step 7. The S-EDA stops its evolution process and returns the best solution found so far when either of the following conditions is satisfied (Step 7): (1) $g = G^{\max}$ and $cnt = CNT^{\max}$, and (2) $|f_{best}^g - f_{best}^g| < \delta_{fit}$ and $Var(P_g) < \delta_{var}$.

3.2 Description of ID²C-EDA

In the authors' previous works [8] and [10], the novel diversity controlling GA and EDA called ID²C-GA and ID²C-EDA were developed based on a real-coded GA (called S-GA) and S-EDA, respectively. In [8], ID²C-GAs and ID²C-EDAs were used for coevolutionary learning in which the objective is to find optimal (P-optimizing) negotiation strategies for interacting agents with incomplete information. Although this work is similar to [8] and [10] in a sense that both adopt EAs for (a similar type of) coevolutionary learning, they are mainly different in two ways: (1) [8] and [10] only focused on finding negotiation strategies for optimizing price only but this work deals with the more difficult problem of finding negotiation strategies that can optimize both price and negotiation speed and (2) while the fitness functions of [8] and [10] are directly related with the (price) utility function, the fitness functions of this work has indirect relationship with the (price and speed) utility functions. It is noted that it has been proved theoretically and demonstrated empirically that the performances of GAs and EDAs are found to be very close to each other although they adopt quite different search strategies [17, 25]. Furthermore, it is empirically observed in [10] that: (1) ID²C-GAs and ID²C-EDAs outperform S-GAs and S-EDAs for the coevolutionary learning because ID²C-GAs and ID²C-EDAs have enough capability for overcoming premature convergence and achieving non-biased coevolution results for both populations and (2) ID²C-EDAs ensures better efficacy and reliability in achieving good solutions than ID²C-GAs if the search space is (very) large. For these reasons, this work adopts ID²C-EDAs and S-EDAs for the coevolutionary learning to carry out comparative studies on their coevolution performance. ID²C-EDAs adopts a subspace-based dynamic (i.e., adaptive) diversity controlling technique called modified (i.e., improved) diversification and refinement (mDR) and two local improvement methods such as population repair (PR) and local neighborhood search (LNS). The pseudocode of ID²C-EDA is presented as follows.

Step 1. Initialization

Generate the initial population P_0 with n_P individuals at random;

$g \leftarrow 0$; $cnt \leftarrow 0$.

Step 2. Selection

$g++$;

Select a set of promising candidates S_{g-1} with $n_S (< n_P)$ individuals from P_{g-1} .

Step 3. Building Model

Estimate the probability distribution $f_{X^g}(x^g)$ from S_{g-1} .

Step 4. Sampling Model

Generate offspring O_g with n_O individuals by sampling $f_{X^g}(x^g)$.

Step 5. Replacement

Create a new population P_g by replacing some individuals of P_{g-1} with O_g .

Step 6. Diversification and Refinement (DR)

Calculate $\text{Div}(P_g)$;

If $\text{Div}(P_g) < \delta_{low}$ and $\text{Div}(P_g) > \delta_{high}$, conduct the following procedures of DR

- A. Pre-ordering individuals in P_g in both fitness and solution spaces;
- B. Eliminating redundant individuals in P_g using the similarity of individuals;
- C. Calculating BOF_i and update C_i ($1 \leq i \leq n_{band}$);
- D. Eliminating infeasible bands if $g > G^{max_infeasible_band}$;
- E. Refining the population using diversified artificial individuals (DAIs)
 - (a) Generating DAIs using BOFs based on population diversity;
 - (b) Injecting the generated DAIs into the population.

Else calculate BOF_i and update C_i ($1 \leq i \leq n_{band}$).

Step 7. Population Repair (PR)

Replace some infeasible individuals consisting of an inappropriate population configuration with new individuals randomly generated using the feasible individual list (FI_List).

Step 8. Local Neighborhood Search (LNS)

Replace some less feasible individuals (having lower fitness) by the neighborhoods generated from the locally best solution in the population.

Step 9. Reinitializing Population and Restarting Evolution

If $cnt < CNT^{max}$ and an inappropriate configuration is detected,

initialize P_g at random;

$g \leftarrow 0$; $cnt++$;

Go to Step 2.

Step 10. Termination

If the termination criteria are not satisfied, go to Step 2.

Else return the best solution found so far.

mDR (Step 6) is the main part of ID²C-EDA and its objective is to achieve individuals (of a population) that are both feasible and diversified through a dynamic diversity control of the population. mDR utilizes robustness of bands (ROBs) in which a band is defined as a distinct (small) fraction of solution space and the solution space is mapped into bands with each fixed size of $Band_Size$. A band i is defined as the more robust one than another band j ($i \neq j$ and $1 \leq i, j \leq n_{band}$) if more individuals belonging to i have survived for more generations than j . For measuring ROBs of solution space at each generation g , each band i : (1) counts band-occupying frequency (BOF_i) which is the accumulated frequency of individuals belonging to i until g is reached and (2) stores BOF_i into the global counter variable C_i . mDR operates selectively depending on the population diversity of P_g , $\text{Div}(P_g)$. mDR operates if $\text{Div}(P_g)$ is below the given lowest possible threshold δ_{low} (i.e., $\text{Div}(P_g) < \delta_{low}$) or is above the given highest possible threshold δ_{high} (i.e., $\text{Div}(P_g) > \delta_{high}$). Otherwise (i.e., $\delta_{low} \leq \text{Div}(P_g) \leq \delta_{high}$), mDR does not operate. In this way, BOF_i is calculated and C_i is updated, $1 \leq i \leq n_{band}$. mDR has two main functionalities: (1) *diversification* ensures achieving a sufficiently high population diversity (from A and B in Step 6), and (2) *refinement* guarantees achieving a refined population consisting of more promising (i.e., feasible and diversified) solutions (from D and E in Step 6) using ROB information (from C in Step 6). The details of mDR are as follows:

A. Pre-ordering the population.

Before eliminating redundant individuals, ordering individuals in P_g is carried out in both the fitness and solution spaces. First, individuals in P_g are sorted according to their fitness values in decreasing order. Next, if there are some individuals with the same fitness values (or less than the predefined threshold δ_{fit}), then the individuals are sorted according to their solution values in decreasing order. Then, the ordered population $P_g^{Ordered}$ is obtained.

B. Eliminating redundant individuals.

Domination of redundant (or duplicate) individuals can lead to premature convergence. This is because redundant individuals with a similar structure reduce population diversity; parents with a similar structure can often reproduce offspring with the same (or very similar) structure in the next generation. To avoid such premature convergence, redundant individuals in $P_g^{Ordered}$ are eliminated from $P_g^{Ordered}$ after testing the similarity among individuals in both fitness and solution levels as follows. If i -th and j -th individuals ($1 \leq i \leq n_P$ and $i+1 \leq j \leq n_P$) in $P_g^{Ordered}$ have very close fitness values (that is less than the given threshold α) and also very close solution values (that is less than the given threshold

β), the j -th individual is considered as the redundant individual and is eliminated from $P_g^{Ordered}$. This similarity test is carried out from $i = 1$ to $n_p - 1$. Then, the population $P_g^{Eliminated}$ with high population diversity can be achieved.

C. Calculating BOFs for all bands.

For each band i ($1 \leq i \leq n_{band}$), BOF_i is calculated from $P_g^{Eliminated}$ and the corresponding C_i is updated. By calculating BOFs from $P_g^{Eliminated}$ —not from P_g or $P_g^{Ordered}$ in which both can have redundant individuals, only the individuals ensuring the higher population diversity will contribute to calculating BOFs. Therefore, reliable BOFs without redundant information are guaranteed.

D. Eliminating infeasible bands.

If C_i of the band i is not updated (i.e., there is no individual belonging to i) during a certain number of generations ($G^{max_infeasible_band}$), i is considered as the infeasible band and is removed from the feasible band list (FB_List) containing all feasible bands. This procedure operates if $g > G^{max_infeasible_band}$ because at least $G^{max_infeasible_band}$ is required to collect infeasible band information for carrying out such elimination.

E. Refining the population.

Since redundant individuals are eliminated (in the B -th Step), diversified artificial individuals (DAIs) are generated and injected into $P_g^{Eliminated}$ equal to the number of the eliminated individuals. For generating feasible DAIs using achieved reliable BOF information, bands belonging to FB_List and having high values of BOFs (i.e., representing high robustness) are considered as promising solution regions for future evolutionary search. As evolution progresses, more effective BOFs will be achieved because more reliable ROB information will be gathered over all bands at later generations.

- (a) Each promising DAI is generated randomly in the selected band based on the two modes: (1) In exploration mode ($\text{Div}(P_g^{Eliminated}) < \delta_{low}$), the bands with the lower BOFs have a higher probability to be selected for generating DAIs and (2) In exploitation mode ($\text{Div}(P_g^{Eliminated}) > \delta_{high}$), the bands with the higher BOFs have a higher probability to be selected for generating DAIs.
- (b) The generated DAIs are injected into $P_g^{Eliminated}$.

Finally, $P_g^{Refined}$ can be ensured that all individuals are both feasible and diversified.

Although the reinitializing population and restarting evolution (in Step 6 of S-EDA and Step 9 of ID²C-EDA) can be a solution for both *Type-I* and the *Type-II* errors resulting in domination of infeasible individuals, extremely large overheads (in terms of both computation and time) are inevitable using the procedure. Therefore, PR and LNS are devised to overcome such drawbacks.

PR (Step 7) is devised to prevent domination of infeasible individuals (having fitness values of all 0s or all 1s) due to the *Type-II* error by replacing infeasible individuals with feasible individuals. Using the feasible individual list (FI_List) consisting of feasible individuals stored in the previous evolution step, PR replaces infeasible individuals with randomly selected individuals from FI_List .

LNS (Step 8) is devised to solve non-convergence or very slow convergence due to the *Type-I* error by compensating degradation of the S-EDA's search efficiency during coevolution process. First, LNS generates effective solution candidates from the neighboring bands of the band involving the current local optimal solution. Then, LNS replaces a small number of individuals (denoted as LNS_SIZE) having the lowest fitness values with the generated solution candidates. The replaced solution candidates can contribute to accelerating convergence of the population.

Since PR and LNS replace some infeasible and less feasible individuals with more promising solutions, they can be considered as replacement techniques. Whereas the replacement (in Step 5) is the technique that simply replaces some individuals of P_{g-1} with some individuals from O_g to create P_g , PR and LNS are adaptive techniques that replace some infeasible and less feasible individuals in P_g (after Step 5) or $P_g^{Refined}$ (after Step 6) with feasible and more promising solution candidates.

4 PS-optimizing agents using acceptability zones

In a complete information setting, any adoption of SS_{PS-opt} can be allowed because it is possible for a PS-optimizing agent to calculate optimal PS-optimizing negotiation outcomes and corresponding negotiation strategies using its opponent's known information. However, in practical situations, it may be difficult for agents to obtain complete information about their opponents because agents generally do not expose their private information, strategies and preferences due to strategic reasons. Nevertheless, one of the greatest challenges in designing agents for PS-optimizing negotiations is the mathematical formulations of optimal PS-optimizing negotiation outcomes and negotiation strategies under a complete information setting (Sect. 4.1). This is because they can be directly used to verify the effectiveness (or correctness) of the coevolved solutions under an incomplete information setting (Sect. 4.2). Herein, PS-optimization under an incomplete information setting largely depends on the choice of SS_{PS-opt} . Hence, for designing PS-optimizing agents operating properly in both complete and incomplete information settings, it is crucial to choose effective SS_{PS-opt} of both PS-optimizing agents.

In determining SS_{PS-opt} , the two solution spaces SS_{NP} and SS_{NS} are considered independently. That is, an agent x

adopts $AccZ_x^{NP}$ as SS_{NP} for optimizing price and $AccZ_x^{NT}$ as SS_{NS} for optimizing negotiation speed. Such SS_{PS-opt} adoption provides a significant advantage for executing (population-based) PS-optimizing negotiations and finding effective negotiation strategies for both PS-optimizing agents B and S using a coevolutionary learning approach under an incomplete information setting (Sect. 4.2). This is because B and S do not require each other's private negotiation parameters for establishing SS_{PS-opt} before conducting a PS-optimizing negotiation as $AccZ_x^{NP}$ and $AccZ_x^{NT}$ of x are independently considered from those of its opponent in the PS-optimization process. To demonstrate the effectiveness of such SS_{PS-opt} , consider a counter-example such that AgZ^{NP} and AgZ^{NT} are adopted as SS_{NP} and SS_{NS} , respectively, for SS_{PS-opt} . Then, before carrying out a PS-optimizing negotiation, each agent needs to achieve its opponent's private information (such as RP and deadline) for establishing SS_{PS-opt} ; however, estimating the opponent's accurate private information is a very difficult (and sometimes impossible) problem itself under an incomplete information setting.

4.1 PS-optimizing agents with complete information

Given w_{NP}^x and w_{NS}^x , each PS-optimizing agent x with complete information is designed to use: (1) $AccZ_x^{NP}$ (i.e., $[IP_x, RP_x]$) for optimizing price using w_{NP}^x and (2) $AccZ_x^{NT}$ (i.e., $[0, \tau_x]$) for optimizing negotiation speed using w_{NS}^x . Although x still adopts the price utility U_{NP}^x in (5), the speed utility U_{NS}^x in (6) needs to be further modified to reflect characteristics of the preference of negotiation speed of x .

Using w_{NP}^x and w_{NS}^x , x decides the desired agreement price (dP_c^x) in $AccZ_x^{NP}$ and desired agreement time (dT_c^x) in $AccZ_x^{NT}$, respectively. First, x determines dP_c^x satisfying the preference criterion of price in $AccZ_x^{NP}$ using w_{NP}^x :

$$dP_c^x = RP_x + \beta \cdot w_{NP}^x \cdot |RP_x - IP_x|$$

where $\beta = 1$ if $x = B$ and $\beta = -1$ if $x = S$. (8)

x treats dP_c^x as the most favorable possible agreement price because at which x maximizes U_{NP}^x in (5) and hence, it is the upper bound of U_{NP}^x satisfying its preference criterion of price; (1) if x concedes less price utility than $U_{NP}^x(dP_c^x)$, the price utility of its opponent will be decreased, and conversely, (2) if x concedes more price utility than $U_{NP}^x(dP_c^x)$, U_{NP}^x will be decreased. Second, x determines dT_c^x satisfying the preference criterion of negotiation speed in $AccZ_x^{NT}$ using w_{NS}^x :

$$dT_c^x = \tau_x \cdot (1 - w_{NS}^x). \tag{9}$$

x treats the range of negotiation time in $[0, dT_c^x]$ as the favorable possible agreement times because: (1) all the agreement

times shorter than dT_c^x satisfy the preference criterion of negotiation speed and (2) when the negotiation begins, all the negotiation times in $[0, dT_c^x]$ can be mutually favorable possible agreement times satisfying the preference criterion of negotiation speed for both x and its opponent. To incorporate U_{NS}^x in (6) with the favorable possible agreement times, we define the new speed utility function $U_{NS-mapped}^x(T_c^x)$ for an agreement time T_c^x as follows:

$$U_{NS-mapped}^x(T_c^x) = \begin{cases} U_{NS}^x(dT_c^x), & \text{if } T_c^x \in [0, dT_c^x], \\ U_{NS}^x(T_c^x), & \text{otherwise.} \end{cases} \tag{10}$$

Then, the total utility function U_{Total}^x in (7) is modified as follows:

$$U_{Total-mapped}^x(P_x, T_x) = w_{NP}^x \times U_{NP}^x(P_x) + w_{NS}^x \times U_{NS-mapped}^x(T_c^x). \tag{11}$$

Compared to P-optimizing agents, each PS-optimizing agent x : (1) makes concessions in the range of prices from IP_x up to the price less than dP_c^x , which corresponds to (at most) the amount of the price utility $w_{NP}^x(|IP_x - dP_c^x|)$, and (2) aims to achieve a faster agreement time that is equal to or less than dT_c^x in the hope of achieving (at least) the amount of speed utility $U_{NS}^x(|\tau_x - dT_c^x|)$. In a P-optimizing negotiation under a complete information setting, P_c^{P-opt} and T_c^{P-opt} can be specified by either of Theorems 1 and 2 (Sect. 2.1) depending on a bargaining advantage in terms of time. For a PS-optimizing negotiation under a complete information setting, a similar analysis based on a bargaining advantage in terms of time can be applied to determine the optimal agreement price and negotiation time.

At first, we define possible AgZ^{NP} and AgZ^{NT} between the PS-optimizing agents B and S in NSS. The possible AgZ^{NP} determined by dP_c^B and dP_c^S is given as $[\min(dP_c^B, dP_c^S), \max(dP_c^B, dP_c^S)]$ and the possible AgZ^{NT} determined by dT_c^B and dT_c^S is given as $[0, \min\{dT_c^B, dT_c^S\}]$ —which is the overlapping region of dT_c^B and dT_c^S . Then, following Definition 2, PS-optimizing agents are designed to optimize price and speed utilities in the possible AgZ^{NP} and AgZ^{NT} , respectively, to maximize the total utility in (11). Next, the optimal PS-optimizing negotiation outcomes consisting of the optimal agreement price (P_c^{PS-opt}) and optimal agreement time (T_c^{PS-opt}) will be determined based on a bargaining advantage in terms of time and are defined as follows.

Definition 3 If a PS-optimizing agent x has a bargaining advantage in terms of time over its opponent, (1) P_c^{PS-opt} is the price that maximizes U_{NP}^x in the possible AgZ^{NP} and (2) T_c^{PS-opt} is the range of negotiation times satisfying both agents' preference criteria of negotiation speed in the possible AgZ^{NT} .

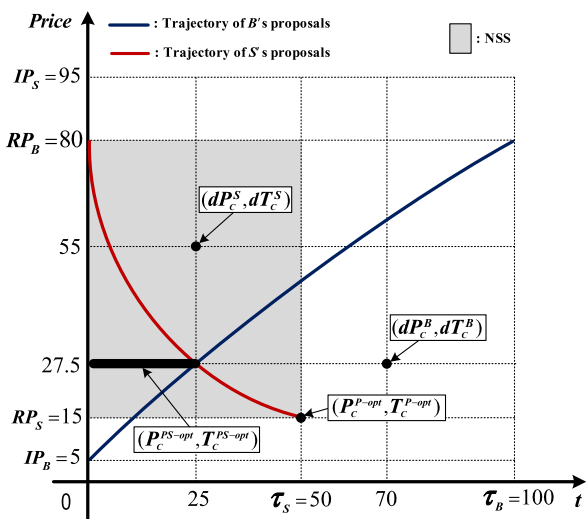


Fig. 2 Agreement behavior of PS-optimizing negotiation when sufficient AgZ^{NP} is provided

Following Definition 3, P_c^{PS-opt} and T_c^{PS-opt} are obtained from the following Theorems 3 and 4 depending on a bargaining advantage in terms of time.

Theorem 3 *If the PS-optimizing agent B has a longer deadline than the PS-optimizing agent S, (1) P_c^{PS-opt} is made at $\min(dP_c^B, dP_c^S)$ and (2) any agreement time in $[0, \min(dT_c^B, dT_c^S)]$ is T_c^{PS-opt} .*

Proof Since B has a longer deadline than S, B has a bargaining advantage over S in terms of time; hence, the final agreement price and agreement time will be completely determined by B. From Definition 3, P_c^{PS-opt} is $\min(dP_c^B, dP_c^S)$ at which U_{NP}^B is maximized. Since $AccZ_B^{NT}$ is $[0, dT_c^B]$ and $AccZ_S^{NT}$ is $[0, dT_c^S]$, T_c^{PS-opt} is determined as $[0, \min\{dT_c^B, dT_c^S\}]$ —which is the overlapping region between $AccZ_B^{NT}$ and $AccZ_S^{NT}$ and at which both B and S satisfy their preference criteria of negotiation speed. □

Theorem 4 *If the PS-optimizing agent S has longer deadline than the PS-optimizing agent B, (1) P_c^{PS-opt} is made at $\max(dP_c^B, dP_c^S)$ and (2) any agreement time in $[0, \min(dT_c^B, dT_c^S)]$ is T_c^{PS-opt} .*

Proof Symmetrically, P_c^{PS-opt} is $\max(dP_c^B, dP_c^S)$ at which U_{NP}^S is maximized; T_c^{PS-opt} is the overlapping region $[0, \min\{dT_c^B, dT_c^S\}]$ at which both B and S satisfy their preference criteria of negotiation speed. □

Figure 2 shows an example of the agreement behavior between PS-optimizing agents B and S when B has a longer deadline than S. The negotiation parameters for B and S are as follows: (1) $IP_B = 5, RP_B = 80, \tau_B = 100$

and $(w_{NP}^B, w_{NS}^B) = (0.7, 0.3)$ for B; (2) $IP_S = 95, RP_S = 15, \tau_S = 50$ and $(w_{NP}^S, w_{NS}^S) = (0.5, 0.5)$ for S. The NSS is $[15, 80]$ for AgZ^{NP} and $[0, 50]$ for AgZ^{NT} . dP_c^x and dT_c^x are determined by (8) and (9), respectively: (1) For the given $(w_{NP}^B, w_{NS}^B) = (0.7, 0.3)$, $dP_c^B = 27.5$ and $dT_c^B = 70$ and (2) for the given $(w_{NP}^S, w_{NS}^S) = (0.5, 0.5)$, $dP_c^S = 55$ and $dT_c^S = 25$. Then, the possible AgZ^{NP} is determined as $[55, 70]$ and the possible AgZ^{NT} is determined as $[0, 25]$. Finally, P_c^{PS-opt} and T_c^{PS-opt} will be determined by Theorem 3 (because B has the bargaining advantage over S); hence, P_c^{PS-opt} is 27.5 and T_c^{PS-opt} is $[0, 25]$.

We have so far determined P_c^{PS-opt} and T_c^{PS-opt} using Theorems 3 and 4 under the assumption that P_c^{PS-opt} and T_c^{PS-opt} belong to NSS (e.g., Fig. 2). However, if P_c^{PS-opt} and T_c^{PS-opt} is outside of NSS, any agreement cannot be made within NSS; therefore, Theorems 3 and 4 can be no longer applied to determine P_c^{PS-opt} and T_c^{PS-opt} . In general, P_c^{PS-opt} and T_c^{PS-opt} sometimes may be outside of NSS depending on the input parameter values of the agent d having a bargaining advantage in terms of time. This is because d determines P_c^{PS-opt} and T_c^{PS-opt} based on optimizing $AccZ_d^{NP}$ and $AccZ_d^{NT}$ using w_{NP}^d and w_{NS}^d , respectively, without considering the agreement zones between d and its opponent (i.e., d does not consider AgZ^{NP} and AgZ^{NT} for carrying out the PS-optimizing negotiation). P_c^{PS-opt} can be made outside of AgZ^{NP} if: (1) d is B and dP_c^B is less than RP_S and (2) d is S and dP_c^S is larger than RP_B (e.g., see Fig. 3). However, in the case of T_c^{PS-opt} , it always belongs to AgZ^{NT} because dT_c^B and dT_c^S are always less than or equal to τ_B and τ_S , respectively. In summary, it is concluded that for x : (1) if sufficient AgZ^{NP} is provided for carrying out a PS-optimizing negotiation, P_c^{PS-opt} and T_c^{PS-opt} can be determined using either Theorems 3 or 4; however, (2) if AgZ^{NP} is provided for carrying out a PS-optimizing negotiation, P_c^{PS-opt} and T_c^{PS-opt} cannot be determined using Theorems 3 and 4.

Given that insufficient AgZ^{NP} is provided for carrying out a PS-optimizing negotiation, the following definition is adopted for designing agreement behaviors of PS-optimizing agents.

Definition 4 *If insufficient AgZ^{NP} is provided for carrying out a PS-optimizing negotiation, P_c^{PS-opt} and T_c^{PS-opt} are made at the nearest point in NSS from dP_c^d and dT_c^d of the agent d having a bargaining advantage in terms of time.*

Definition 4 leads to Theorem 5 showing that the proposed PS-optimizing agents satisfy (at least) the minimum performance requirement for PS-optimizing agents (defined in Sect. 2.2) even under the given condition that insufficient AgZ^{NP} is provided.

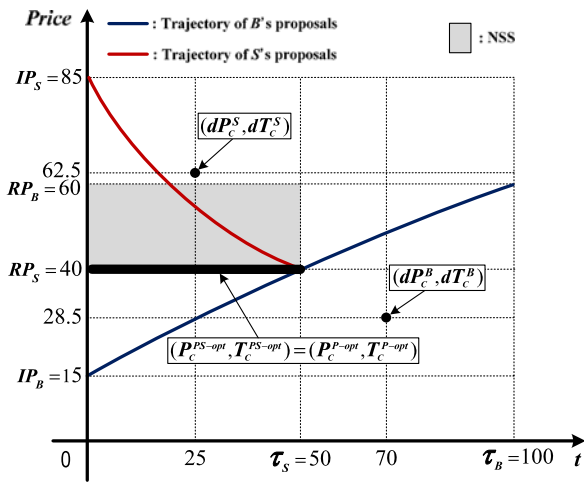


Fig. 3 Agreement behavior of PS-optimizing negotiation when insufficient AgZ^{NP} is provided

Theorem 5 *If insufficient AgZ^{NP} is provided, PS-optimizing negotiation outcomes equal to P-optimizing negotiation outcomes for the given same negotiation settings.*

Proof If B has a bargaining advantage in terms of time, the nearest price from dP_c^B in NSS is RP_S (at which U_{NP}^B is maximized) and the nearest negotiation time is τ_S . Hence, following Definition 4, P_c^{PS-opt} and T_c^{PS-opt} are made at RP_S and τ_S , respectively—which has the same result of the P-optimizing negotiation given by Theorem 1. Similarly, if S has a bargaining advantage in terms of time, the nearest price from dP_c^S in NSS is RP_B (at which the price utility U_{NP}^S is maximized) and the nearest negotiation time is τ_B . Hence, following Definition 4, P_c^{PS-opt} and T_c^{PS-opt} are made at RP_B and τ_B , respectively—which has the same result of the P-optimizing negotiation given by Theorem 2. \square

Figure 3 shows an example of the agreement behaviors between PS-optimizing agents B and S when (1) B has a longer deadline than S and (2) insufficient AgZ^{NP} is provided. The negotiation parameters for B and S are as follows: (1) $IP_B = 15, RP_B = 60, \tau_B = 100$ and $(w_{NP}^B, w_{NS}^B) = (0.7, 0.3)$ for B ; (2) $IP_S = 85, RP_S = 40, \tau_S = 50$ and $(w_{NP}^S, w_{NS}^S) = (0.5, 0.5)$ for S . The NSS is $[40, 60]$ for AgZ^{NP} and $[0, 50]$ for AgZ^{NT} . dP_c^x and dT_c^x are determined by (8) and (9), respectively; (1) For the given $(w_{NP}^B, w_{NS}^B) = (0.7, 0.3)$, $dP_c^B = 28.5$ and $dT_c^B = 70$ and (2) for the given $(w_{NP}^S, w_{NS}^S) = (0.5, 0.5)$, $dP_c^S = 62.5$ and $dT_c^S = 25$. Then, the possible AgZ^{NP} is determined as $[40, 60]$ and the possible AgZ^{NT} is determined as $[0, 25]$. Since B has a bargaining advantage over S in terms of time, P_c^{PS-opt} is 28.5 and T_c^{PS-opt} is $[0, 25]$ following Theorem 3. However, P_c^{PS-opt} is not in AgZ^{NP} while T_c^{PS-opt} is in AgZ^{NT} ; the agreement points of such PS-optimizing negotiation are not made in

NSS (i.e., PS-optimizing negotiation fails without making an agreement). Therefore, from Theorem 5, P_c^{PS-opt} and T_c^{PS-opt} are made at 28.5 and 25, respectively—which are equal to P_c^{P-opt} and T_c^{P-opt} achieved from Theorem 1 (i.e., $P_c^{P-opt} = 28.5$ and $T_c^{P-opt} = 25$).

Finally, from the achieved P_c^{PS-opt} and T_c^{PS-opt} (using one of Theorems 3 to 5), optimal negotiation strategies of PS-optimizing agents B and S (to carry out the optimal PS-optimizing negotiation) can be achieved. Specifically, the optimal PS-optimizing negotiation strategies of B and S are derived from (1) by substituting P_t^x and t in (1) by P_c^{PS-opt} and T_c^{PS-opt} , respectively, as follows:

$$\lambda_B^{PS-opt} \leq \log_{\left(\frac{T_c^{PS-opt}}{\tau_B}\right)} \left(\frac{P_c^{PS-opt} - IP_B}{RP_B - IP_B} \right), \tag{12}$$

$$\lambda_S^{PS-opt} \leq \log_{\left(\frac{T_c^{PS-opt}}{\tau_S}\right)} \left(\frac{IP_S - P_c^{PS-opt}}{IP_S - RP_S} \right). \tag{13}$$

4.2 PS-optimizing agents with incomplete information

Given w_{NP}^x and w_{NS}^x , each PS-optimizing agent x with incomplete information also adopts: (1) $AccZ_x^{NP}$ for optimizing price using w_{NP}^x and (2) $AccZ_x^{NT}$ for optimizing negotiation speed using w_{NS}^x . Since PS-optimizing agents with incomplete information do not know their opponents' private information (such as RP , deadline and preferences for price and negotiation time), their opponents' desired agreement points are unknown. Therefore, the agents cannot apply one of Theorems 3 to 5 directly for determining their optimal negotiation outcomes. Owing to the lack of information about their opponents' private information, this research adopts a coevolutionary learning approach to find effective PS-optimizing negotiation strategies for both PS-optimizing agents B and S . Coevolutionary learning approaches have long been used to model competitive coevolution problems (e.g., the iterated prisoner's dilemma [3]).

Given that PS-optimizing agents B (with IP_B, RP_B, τ_B and (w_{NP}^B, w_{NS}^B)) and S (with IP_S, RP_S, τ_S and (w_{NP}^S, w_{NS}^S)), the following three key components are required for the coevolutionary learning:

- (C1) *Creating populations:* Two heterogeneous populations with size n_P are created: POP_B where individuals consist of B s and POP_S where individuals consist of S s. Throughout the rest of this paper, we use: (1) the term “individual” interchangeably with “agent” and (2) the term “solution” interchangeably with “(PS-optimizing) negotiation strategy” depending on the context.
- (C2) *Initialization:* Individuals of POP_B and POP_S are initialized. All individuals in POP_B are initialized with IP_B, RP_B, τ_B and (w_{NP}^B, w_{NS}^B) ; however, the

negotiation strategy of each individual in POP_B is randomly determined in the possible strategy range $[\lambda_{lower}, \lambda_{upper}]$ where λ_{lower} is the lower bound of possible strategies and λ_{upper} is the upper bound of possible strategies. In the same manner, all individuals in POP_S are initialized with IP_S, RP_S, τ_S and (w_{NP}^S, w_{NS}^S) ; however, the negotiation strategy of each individual in POP_S is randomly determined in $[\lambda_{lower}, \lambda_{upper}]$. Hence, individuals are mainly characterized by their negotiation strategies in its population.

(C3) *Making interactions between populations:* Coevolutionary interactions between POP_B and POP_S are carried out as follows:

- a. Individuals in POP_B and POP_S are randomly chosen and matched in a one-to-one manner.
- b. Each matched pair of POP_B and POP_S conducts a PS-optimizing negotiation without the knowledge of private information of its opponent.

As a result of the coevolutionary interaction, individuals in POP_B and POP_S obtain negotiation outcomes.

The coevolutionary learning procedure using the same type of EDAs for both POP_B and POP_S is as follows. Two EDAs (i.e., either two S-EDAs or two ID²C-EDAs) are adopted for coevolving POP_B and POP_S , respectively: one EDA for POP_B and the other EDA for POP_S . Here, Step 1 of S-EDAs and ID²C-EDAs is substituted by the above C1 and C2 to create populations and initialize them. Throughout its evolution procedure described in Sect. 3, each EDA evolves solutions (of individuals) in its population. Each EDA evaluates fitness of individuals from the negotiation outcomes of individuals obtained from C3. Therefore, C3 is executed in the fitness evaluation stage before applying selection. In summary, the negotiation strategies of agents in both POP_B and POP_S are coevolved from: (1) the coevolutionary interaction between POP_B and POP_S in C3 and (2) the evolution procedure for evolving solutions in each population.

In the coevolutionary learning procedure, the main issues for finding (or coevolving) effective PS-optimizing negotiation strategies of B and S are: (1) adopting (or developing) EAs suitable for the coevolution, and (2) given (w_{NP}^x, w_{NS}^x) , designing an appropriate fitness function that can achieve good candidate solutions for a PS-optimizing negotiation.

First, for coevolving optimal PS-optimizing strategies between B and S , special EAs called ID²C-EDAs were adopted for both POP_B and POP_S due to its effectiveness in coevolution learning [8, 10] while S-EDAs were used for comparative studies of the coevolution performance. The coevolutionary learning approach adopting ID²C-EDAs allows us to achieve an approximation to the optimal PS-optimizing negotiation strategies achieved in the complete information setting in Sect. 4.1.

We then need to consider how a PS-optimizing agent x represents the preference of price using w_{NP}^x in $AccZ_x^{NP}$ and the preference of negotiation speed using w_{NS}^x in $AccZ_x^{NT}$ in an incomplete information setting. The same definitions of dP_c^x in (8) and dT_c^x in (9) were adopted for x with incomplete information adopts. Accordingly, x with incomplete information treats dP_c^x as the most favorable agreement price and dT_c^x as one of the favorable agreement times. Then, the fitness function maximizing its values at dP_c^x and dT_c^x for the given (w_{NP}^x, w_{NS}^x) is required. We will briefly examine some drawbacks of the fitness functions in the previous studies and then describe the details of the proposed fitness function.

In most of the previous studies (e.g., [8–10, 23, 32] and [19] where they mostly dealt with P-optimizing negotiations), fitness functions as the form of utility functions were widely adopted. Therefore, total utility functions U_{Total}^x in (7) and $U_{Total-mapped}^x$ in (11) also can be adopted as fitness functions for the coevolutionary learning. To do this, we need to evaluate the suitability (or effectiveness) of the fitness functions adopting U_{Total}^x and $U_{Total-mapped}^x$. U_{Total}^x is calculated by linearly combining the results of: (1) U_{NP}^x in (5) multiplied by w_{NP}^x and (2) U_{NS}^x in (6) multiplied by w_{NS}^x . Similarly, $U_{Total-mapped}^x$ is calculated by linearly combining the results of: (1) U_{NP}^x in (5) multiplied by w_{NP}^x and (2) $U_{NS-mapped}^x$ in (10) multiplied by w_{NS}^x . The difference between the fitness functions using U_{Total}^x and $U_{Total-mapped}^x$ is that the fitness function adopting U_{Total}^x considers the different emphases in calculating U_{NS}^x in $[0, dT_c^x]$ by giving a higher value to U_{NS}^x for a smaller agreement time. However, the fitness function adopting $U_{Total-mapped}^x$ has the same value of U_{NS}^x for all agreement times belonging to $[0, dT_c^x]$ as $U_{NS}^x(dT_c^x)$. Using the fitness function as either U_{Total}^x or $U_{Total-mapped}^x$, the coevolution performance in terms of intensification capability for coevolving converged solutions can be severely deteriorated. As a result, both EDAs for POP_B and POP_S generally cannot evolve effective PS-optimizing negotiation strategies (within reasonable generations). In the case of the fitness function adopting U_{Total}^x , this is mainly because the fitness at different P_c^x and T_c^x can have the same value as $U_{Total}^x(dP_c^x, dT_c^x)$. For example, consider the case that: (1) an individual i obtained a negotiation outcome at the agreement price P_c^i and the agreement time T_c^i where U_T^i consists of $U_{NP}^i(P_c^i)$ ($< U_{NP}^i(dP_c^i)$) and $U_{NS}^i(P_c^i)$ ($> U_{NS}^i(dT_c^i)$) and (2) another individual j obtained a negotiation outcome at the agreement price P_c^j and the agreement time T_c^j where U_{Total}^j consists of $U_{NP}^j(P_c^j)$ ($> U_{NP}^j(dP_c^j)$) and $U_{NS}^j(T_c^j)$ ($< U_{NS}^j(dT_c^j)$). Then, there are many possible combinations of values for (P_c^i, T_c^i) and (P_c^j, T_c^j) where the fitness $fit(P_c^i, T_c^i)$ —set as $U_{Total}^i(P_c^i, T_c^i) = w_{NP}^i \times U_{NP}^i(P_c^i) + w_{NS}^i \times U_{NS}^i(T_c^i)$ —is equal to $fit(P_c^j, T_c^j)$ —set as $U_{Total}^j(P_c^j, T_c^j) = w_{NP}^j \times U_{NP}^j(P_c^j) + w_{NS}^j \times U_{NS}^j(T_c^j)$, in which both are equal to

$U_{Total}^x(dP_c^x, dT_c^x)$. Hence, the ambiguity (in representing $fit(P_c^x, T_c^x)$ that needs to be maximized at $U_{Total}^x(dP_c^x, dT_c^x)$) makes the coupled fitness landscape for the coevolutionary learning more complicated (to evolve effective solutions), which leads to the deterioration of intensification capability of both EDAs. In the case of the fitness function adopting $U_{Total-mapped}^x$, the problem of such ambiguity may be solved to some extent compared to the fitness function adopting U_{Total}^x . This is because fitness values can be solely determined by price utility if the agreement times of all individuals are made in $[0, dT_c^x]$ in which speed utilities of all individuals are the same. However, we cannot always guarantee that all agreement times are within $[0, dT_c^x]$ because these depend on the negotiation parameter settings and the evolution of EDAs. Hence, although the fitness function using $U_{Total-mapped}^x$ is more robust than the fitness function adopting U_{Total}^x in charactering more promising solutions, both cannot solve the problem of the ambiguity (completely) and prevent deterioration of intensification capability. From this analysis, we found that the fitness functions (as the form of the total utility functions in U_{Total}^x and $U_{Total-mapped}^x$) are not appropriate for coevolving effective PS-optimizing negotiation strategies (within reasonable numbers of generations).

For designing an effective fitness function, this work uses price and speed likelihood functions instead of using price and speed utility functions directly. Given w_{NP}^x and w_{NS}^x , the price likelihood function (Lh_{NP}^x) measures the likelihood between P_x and dP_c^x and the speed likelihood function (Lh_{NS}^x) measures the likelihood between T_x and dT_c^x .

Lh_{NP}^x for a price P_x is defined as follows:

$$Lh_{NP}^x(P_x) = \begin{cases} \frac{1}{\sqrt{\pi} \cdot \rho_{NP}} \exp\left(-\frac{|RP_x - P_x|}{|RP_x - IP_x| - w_{NP}^x} \cdot \frac{w_{NP}^x}{\rho_{NP}}\right)^2, & \text{if an agreement is reached,} \\ 0, & \text{otherwise,} \end{cases} \quad (14)$$

where $|RP_x - P_x|/|RP_x - IP_x|$ is a relative position of P_x in $AccZ_x^{NP}$. As P_x is close to dP_c^x , $|RP_x - P_x|/|RP_x - IP_x| - w_{NP}^x$ will have a smaller value; hence, a higher value of Lh_{NP}^x will be obtained. Through empirical studies, the deviation of Lh_{NP}^x (i.e., the shape of Lh_{NP}^x) is designed to be very narrow by normalizing it with $\rho_{NP} = w_{NP}^x/100$ in which ρ_{NP} can be considered as a weighting factor to put more emphasis on P_x (i.e., for achieving a higher value of Lh_{NP}^x) if it is closer to dP_c^x . Instead of using price utility in (4), (14) measures the closeness of P_x from dP_c^x using the Gaussian likelihood function designed to maximize $Lh_{NP}^x(P_x)$ at $P_x = dP_c^x$.

Lh_{NS}^x for a negotiation time T_x is defined as follows:

$$Lh_{NS}^x(T_x) = \begin{cases} \frac{1}{\sqrt{\pi} \cdot \rho_{NS}} \exp\left(-\frac{(1.0 - \frac{T_x}{\tau_x}) - w_{NS}^x}{\rho_{NS}}\right)^2, & \text{if an agreement is reached,} \\ 0, & \text{otherwise,} \end{cases} \quad (15)$$

where $(1.0 - T_x/\tau_x)$ is a relative position of T_x in $AccZ_x^{NT}$. As T_x is close to dT_c^x , $(1.0 - T_x/\tau_x) - w_{NS}^x$ will have the smaller value; hence, a higher value of Lh_{NS}^x will be obtained. Through empirical studies, the deviation of Lh_{NS}^x is designed to be not very narrow (compared to the deviation of Lh_{NP}^x) by normalizing it with $\rho_{NS} = w_{NS}^x$. ρ_{NS} has a large value to put less emphasis on dT_c^x because a shorter agreement time will be better for both B and S ; however, setting ρ_{NS} to be too large will slow down the coevolution speed because large possible solution candidates can slow down coevolution speed, which affects the intensification capability of EDAs. Instead of using speed utility in (5) or (10), (15) measures the closeness of T_x from dT_c^x using the Gaussian likelihood function designed to maximize $Lh_{NS}^x(T_x)$ at $T_x = dT_c^x$. In regard to optimizing negotiation speed, a special mapping function $f_{t-map}^x(T_c^x)$ for negotiation time was adopted and is defined as follows:

$$f_{t-map}^x(T_c^x) = \begin{cases} dT_c^x, & \text{if } T_c^x \in [0, dT_c^x], \\ T_c^x, & \text{otherwise.} \end{cases} \quad (16)$$

This is essential to assist and realize the coevolution of effective PS-optimizing negotiation strategies for both B and S because: (1) all T_c^x in $[0, dT_c^x]$ satisfy the preference of negotiation speed and (2) in general, a shorter negotiation time will be better than a longer negotiation time.

Finally, using Lh_{NP}^x in (14) and Lh_{NS}^x in (15) together with f_{t-map}^x in (16), the final proposed fitness function for EDAs is defined as follows:

$$\begin{aligned} fit(P_c^x, T_c^x) &= w_{NP}^x \cdot \exp\left(\frac{|Lh_{NP}^x(P_c^x) - Lh_{NP}^x(dP_c^x)|}{Lh_{NP}^x(dP_c^x)}\right)^{-(1-w_{NP}^x)} \\ &\quad + w_{NS}^x \\ &\quad \cdot \exp\left(\frac{|Lh_{NS}^x(f_{t-map}^x(T_c^x)) - Lh_{NS}^x(dT_c^x)|}{Lh_{NS}^x(dT_c^x)}\right)^{-(1-w_{NS}^x)}. \end{aligned} \quad (17)$$

The more $Lh_{NP}^x(P_c^x)$ is close to $Lh_{NP}^x(dP_c^x)$ and w_{NP}^x is large, the more the value of the exponential function for price in (17) is large. Similarly, the more $Lh_{NS}^x(T_c^x)$ is close to $Lh_{NS}^x(dT_c^x)$ and w_{NS}^x is large, the more the value of the exponential function for negotiation speed in (17) is large. Therefore, $fit(P_c^x, T_c^x)$ emphasizes the exponential functions for price and negotiation speed by linearly combining them with w_{NP}^x and w_{NS}^x , respectively.

5 Empirical evaluation and analysis

In this section, we first detail the methodology for analyzing the performance of coevolved PS-optimizing negotia-

Table 2 Parameter setting for negotiation agents B and S

Parameters		Values	
For Type-I experiments	Possible price range of B	$[IP_B, RP_B]$	[5, 80]
	Possible price range of S	$[IP_S, RP_S]$	[95, 15]
For Type-II experiments	Possible price range of B	$[IP_B, RP_B]$	[15, 60]
	Possible price range of S	$[IP_S, RP_S]$	[85, 40]
For both Type-I and Type-II experiments	Possible strategy range	$[\lambda_{lower}, \lambda_{upper}]$	[0, 5]
	Deadline range	Long	[16, 30]
		Mid	[31, 60]
Short		[61, 120]	

tion strategies (of PS-optimizing agents B and S) using S-EDAs or ID²C-EDAs under an incomplete information setting. We then proceed to the actual empirical study of the coevolved PS-optimizing negotiation strategies by comparing them with the optimal negotiation strategies obtained under a complete information setting. Furthermore, we compare and analyze the empirical results of S-EDAs and ID²C-EDAs obtained from coevolutionary learning.

5.1 Methodology

For PS-optimizing agents B and S , optimal PS-optimizing negotiation outcomes and the corresponding negotiation strategies are subject to the size of AgZ^{NP} as described in Sect. 4.1. Hence, two groups of experiments were designed to evaluate the performance of PS-optimizing negotiations for the different negotiation situations: (1) Type-I experiments for the negotiation settings with sufficient AgZ^{NP} , and (2) Type-II experiments for the negotiations with insufficient AgZ^{NP} . In each group of experiments, the coevolution results using ID²C-EDAs are compared with those of S-EDAs to evaluate their coevolution performance in finding effective PS-optimizing negotiation strategies.

5.1.1 Testbed

To evaluate the performance of the coevolved PS-optimizing negotiation strategies of the proposed PS-optimizing agents, a simulation testbed consisting of a virtual negotiation environment for supporting population-based negotiations in an incomplete information setting using EDAs was implemented using C++. Both POP_B and POP_S were evolved using either S-EDAs or ID²C-EDAs as described in Sect. 3. Coevolutionary interaction is achieved from the (population-based) negotiations between POP_B and POP_S as described in Sect. 4.2. In addition, each POP_B and POP_S has a controller that: (1) generates agents and initializes their negotiation parameters (such as preferences of price and negotiation speed, IP , RP , deadlines and negotiation strategies), (2) manages the information of the matched pairs of agents

between POP_B and POP_S , (3) monitors the termination status of its EDA and shares the information with the controller of the opponent population to check the termination conditions for the coevolution, which is to terminate both EDAs simultaneously, (4) synchronizes PS-optimizing negotiations and handles message passing and payment transfer between all matched agents, and (5) reinitializes its population and restarts evolution of EDAs when the CNT^{max} is reached.

The experiments were conducted on a computer with Windows XP (32-bit) service pack 3, Intel® Core™2 Duo CPU E8500 @ 3.16 GHz & 3.17 GHz and 4 GB RAM.

5.1.2 Experimental settings

The input parameters for two types of PS-optimizing negotiation agents B and S are described in Table 2.

The price ranges (determined by IP s and RP s) and strategy ranges for B and S were adopted for the purpose of the Type-I and Type-II experiments. The deadline range of agents was grouped into three categories empirically: *Long* when negotiation rounds are in [16, 30]; *Moderate* (denoted as *Mid* in Table 2) when negotiation rounds are in [31, 60]; and *Short* when negotiation rounds are in [61, 120]. The deadline ranges [1, 15] and [121, ∞] were not considered because repeated experimental tuning showed that the average success rate of PS-optimizing negotiations is very low when the agents adopt their deadlines in the ranges. Due to space limitation, three representative values from the three categories are chosen, respectively: 20 for Short, 50 for Mid, and 100 for Long. Based on a bargaining advantage in terms of time, there exist six representative deadline combinations between B and S as follows:

- (1) (Long, Mid), (Mid, Short) and (Long, Short) for the case that B has a longer deadline than S ,
- (2) (Mid, Long), (Short, Mid) and (Short, Long) for the case that S has a longer deadline than B .

However, since cases 1 and 2 are symmetrical and the similar analysis can be applied to both cases, we only describe the results for the case 1. In the negotiations between B

Table 3 Combinations of preference criteria of B and S

Optimization mode	Preference criteria of B	Preference criteria of S
Mode 1: <i>more-P-optimizing vs. more-P-optimizing</i>	$(w_{NP}^B, w_{NS}^B) = (0.7, 0.3)$	$(w_{NP}^S, w_{NS}^S) = (0.7, 0.3)$
Mode 2: <i>more-P-optimizing vs. exact-PS-optimizing</i>	$(w_{NP}^B, w_{NS}^B) = (0.7, 0.3)$	$(w_{NP}^S, w_{NS}^S) = (0.5, 0.5)$
Mode 3: <i>more-P-optimizing vs. more-S-optimizing</i>	$(w_{NP}^B, w_{NS}^B) = (0.7, 0.3)$	$(w_{NP}^S, w_{NS}^S) = (0.3, 0.7)$
Mode 4: <i>exact-PS-optimizing vs. more-P-optimizing</i>	$(w_{NP}^B, w_{NS}^B) = (0.5, 0.5)$	$(w_{NP}^S, w_{NS}^S) = (0.7, 0.3)$
Mode 5: <i>exact-PS-optimizing vs. exact-PS-optimizing</i>	$(w_{NP}^B, w_{NS}^B) = (0.5, 0.5)$	$(w_{NP}^S, w_{NS}^S) = (0.5, 0.5)$
Mode 6: <i>exact-PS-optimizing vs. more-S-optimizing</i>	$(w_{NP}^B, w_{NS}^B) = (0.5, 0.5)$	$(w_{NP}^S, w_{NS}^S) = (0.3, 0.7)$
Mode 7: <i>more-S-optimizing vs. more-P-optimizing</i>	$(w_{NP}^B, w_{NS}^B) = (0.3, 0.7)$	$(w_{NP}^S, w_{NS}^S) = (0.7, 0.3)$
Mode 8: <i>more-S-optimizing vs. exact-PS-optimizing</i>	$(w_{NP}^B, w_{NS}^B) = (0.3, 0.7)$	$(w_{NP}^S, w_{NS}^S) = (0.5, 0.5)$
Mode 9: <i>more-S-optimizing vs. more-S-optimizing</i>	$(w_{NP}^B, w_{NS}^B) = (0.3, 0.7)$	$(w_{NP}^S, w_{NS}^S) = (0.3, 0.7)$

Table 4 Parameter settings for S-EDAs and ID²C-EDAs

Type	Parameters	Values
Both S-EDAs and ID ² C-EDAs	n_P	50
	n_S	25
	n_O	50
	G^{\max}	2500
	CNT^{\max}	10
ID ² C-EDA	δ_{fit}	10^{-5}
	α	10^{-3}
	β	10^{-4}
	<i>Band_Size</i>	0.1
	$G^{\max}_{infeasible_band}$	10
S-EDA	δ_{var}	1.35
ID ² C-EDA	δ_{high}	0.125
	δ_{low}	0.05
	δ_{var}	0.0005
ID ² C-EDA	<i>LNS_SIZE</i>	8

and S , the experiments were set such that S starts its negotiation first by proposing its first proposal to B .

Different emphases on price and negotiation speed (i.e., different weightings between w_p and w_s) lead to different groups of preference criteria. Each PS-optimizing agent x has three representative preference criteria as follows:

- (1) *more-P-optimizing case:* $(w_{NP}^x, w_{NS}^x) = (0.7, 0.3)$
- (2) *exact-PS-optimizing case:* $(w_{NP}^x, w_{NS}^x) = (0.5, 0.5)$
- (3) *more-S-optimizing case:* $(w_{NP}^x, w_{NS}^x) = (0.3, 0.7)$

Hence, the following nine combinations are possible between B and S as described in Table 3.

The experimental parameter settings for S-EDAs and ID²C-EDAs are described in Table 4. We used the experimentally tuned parameters from [8] and [10].

5.1.3 Description of results

Even though extensive simulations were carried out for all the situations, only representative results are presented in this section due to space limitation. Empirical results for the Type-I and Type-II experiments are shown in Tables 6 to 8 and Tables 9 to 11, respectively. All the values in the experimental tables were averaged based on more than 10^3 runs. The symbols for the results in Tables 6 to 11 and their descriptions are summarized in Table 5. In the results of Tables 6 to 11, the rows for the performance measures (see Sect. 5.1.4) are shaded; in addition, the rows for the results achieved from ID²C-EDAs are in boldface to discriminate and emphasize them with those achieved from S-EDAs.

5.1.4 Performance measure

Under a complete information setting, optimal PS-optimizing negotiation outcomes and negotiation strategies are achieved as follows: (1) P_c^{PS-opt} and T_c^{PS-opt} are obtained from equilibrium analyses using Theorems 3 to 5 and (2) from the obtained P_c^{PS-opt} and T_c^{PS-opt} , λ_B^{PS-opt} and λ_S^{PS-opt} are calculated using (12) and (13), respectively.

Under an incomplete information setting, if EDAs carry out balanced coevolution (for both POP_B and POP_S), the agreement price and agreement time obtained for B will be very close to the agreement price and agreement time obtained for S , respectively. This is because the optimal agreement price and agreement time of B should be equal to those of S , respectively. To verify the effectiveness of the coevolved PS-optimizing negotiation strategies of both B and S , we compare the coevolution results (obtained from coevolutionary learning using either S-EDAs or ID²C-EDAs) with the optimal results (obtained under a complete infor-

Table 5 Summary of notation for the results

Type	Notation	Description
Agreement Price		
P_c	P_c^{PS-opt}	The optimal agreement price achieved from one of Theorems 3 to 5 under a complete information setting
	$\bar{P}_c^B (S-EDA)$	$\sum P_c^B (S-EDA) / \text{No. of experimental runs}$: Average value of the agreement prices for POP_B achieved from coevolutionary learning using S-EDA for B
	$\bar{P}_c^B (ID^2C-EDA)$	$\sum P_c^B (ID^2C-EDA) / \text{No. of experimental runs}$: Average value of the agreement prices for POP_B achieved from coevolutionary learning using ID ² C-EDA for B
	$\bar{P}_c^S (S-EDA)$	$\sum P_c^S (S-EDA) / \text{No. of experimental runs}$: Average value of the agreement prices for POP_S achieved from coevolutionary learning using S-EDA for S
	$\bar{P}_c^S (ID^2C-EDA)$	$\sum P_c^S (ID^2C-EDA) / \text{No. of experimental runs}$: Average value of the agreement prices for POP_S achieved from coevolutionary learning using ID ² C-EDA for S
Agreement Time		
T_c	T_c^{PS-opt}	The optimal agreement time achieved from one of Theorems 3 to 5 under a complete information setting
	$\bar{T}_c^B (S-EDA)$	$\sum T_c^B (S-EDA) / \text{No. of experimental runs}$: Average value of the agreement times for POP_B achieved from coevolutionary learning using S-EDA for B
	$\bar{T}_c^B (ID^2C-EDA)$	$\sum T_c^B (ID^2C-EDA) / \text{No. of experimental runs}$: Average value of the agreement times for POP_B achieved from coevolutionary learning using ID ² C-EDA for B
	$\bar{T}_c^S (S-EDA)$	$\sum T_c^S (S-EDA) / \text{No. of experimental runs}$: Average value of the agreement times for POP_S achieved from coevolutionary learning using S-EDA for S
	$\bar{T}_c^S (ID^2C-EDA)$	$\sum T_c^S (ID^2C-EDA) / \text{No. of experimental runs}$: Average value of the agreement times for POP_S achieved from coevolutionary learning using ID ² C-EDA for S
Strategy		
λ_B	λ_B^{PS-opt}	PS-optimizing negotiation strategy of B achieved from (13) under a complete information setting
	$\bar{\lambda}_B^{PS-opt} (S-EDA)$	$\sum \lambda_B^{PS-opt} (S-EDA) / \text{No. of experimental runs}$: Average value of the coevolved PS-optimizing negotiation strategies using S-EDA for B
	$\bar{\lambda}_B^{PS-opt} (ID^2C-EDA)$	$\sum \lambda_B^{PS-opt} (ID^2C-EDA) / \text{No. of experimental runs}$: Average value of the coevolved PS-optimizing negotiation strategies using ID ² C-EDA for B
λ_S	λ_S^{PS-opt}	PS-optimizing negotiation strategy of S achieved from (14) under a complete information setting
	$\bar{\lambda}_S^{PS-opt} (S-EDA)$	$\sum \lambda_S^{PS-opt} (S-EDA) / \text{No. of experimental runs}$: Average value of the coevolved PS-optimizing negotiation strategies using S-EDA for S
	$\bar{\lambda}_S^{PS-opt} (ID^2C-EDA)$	$\sum \lambda_S^{PS-opt} (ID^2C-EDA) / \text{No. of experimental runs}$: Average value of the coevolved PS-optimizing negotiation strategies using ID ² C-EDA for S
Convergence of EDA		
N^{Gen}	$\bar{N}^{Gen} (S-EDA)$	$\sum N^{Gen} (S-EDA) / \text{No. of experimental runs}$: Average value of the number of generations for coevolutionary learning using S-EDAs
	$\bar{N}^{Gen} (ID^2C-EDA)$	$\sum N^{Gen} (ID^2C-EDA) / \text{No. of experimental runs}$: Average value of the number of generations for coevolutionary learning using ID ² C-EDAs
N^{Re_init}	$\bar{N}^{Re_init} (S-EDA)$	$\sum N^{Re_init} (S-EDA) / \text{No. of experimental runs}$: Average value of the number of evolution restarts for coevolutionary learning using S-EDAs
	$\bar{N}^{Re_init} (ID^2C-EDA)$	$\sum N^{Re_init} (ID^2C-EDA) / \text{No. of experimental runs}$: Average value of the number of evolution restarts for coevolutionary learning using ID ² C-EDAs

mation setting) by examining the following two conditions: (1) *closeness to the optimum*: the obtained PS-optimizing negotiation outcomes and coevolved PS-optimizing negotiation strategies should be close to the optimal results and (2) *balanced coevolution*: the obtained negotiation outcomes

of B should be the same as those of S as a result of coevolutionary learning.

First, for measuring closeness between the coevolution results and optimal results, the following three types of closeness metric are devised for each type of EDA:

- (1) $\delta_{dist}^{P_c^x}$ measures the closeness between P_c^{PS-opt} and the agreement price \bar{P}_c^x (S-EDA) (respectively, \bar{P}_c^x (ID²C-EDA)) obtained from coevolutionary learning as follows:
 $\delta_{dist}^{P_c^x}$ (S-EDA) = $|\bar{P}_c^x$ (S-EDA) - P_c^{PS-opt} | for the coevolution using S-EDA,
 $\delta_{dist}^{P_c^x}$ (ID²C-EDA) = $|\bar{P}_c^x$ (ID²C-EDA) - P_c^{PS-opt} | for the coevolution using ID²C-EDA

where if S-EDA (respectively, ID²C-EDA) has obtained \bar{P}_c^x (S-EDA) (respectively, \bar{P}_c^x (ID²C-EDA)) that is the same as P_c^{PS-opt} from coevolutionary learning, then $\delta_{dist}^{P_c^x}$ (S-EDA) (respectively, $\delta_{dist}^{P_c^x}$ (ID²C-EDA)) will be 0; otherwise, $\delta_{dist}^{P_c^x}$ (S-EDA) (respectively, $\delta_{dist}^{P_c^x}$ (ID²C-EDA)) will be larger than 0.

- (2) $\delta_{dist}^{T_c^x}$ measures the closeness between T_c^{PS-opt} and the agreement time \bar{T}_c^x (S-EDA) (respectively, \bar{T}_c^x (ID²C-EDA)) obtained from coevolutionary learning as follows:
 $\delta_{dist}^{T_c^x}$ (S-EDA) = \bar{T}_c^x (S-EDA) - $\max(T_c^{PS-opt})$ for the coevolution using S-EDA,
 $\delta_{dist}^{T_c^x}$ (ID²C-EDA) = \bar{T}_c^x (ID²C-EDA) - $\max(T_c^{PS-opt})$ for the coevolution using ID²C-EDA

where we consider its maximum value $\max(T_c^{PS-opt})$ as the basis of closeness because T_c^{PS-opt} can be represented as a range of negotiation time. Hence, if S-EDA (respectively, ID²C-EDA) has obtained \bar{T}_c^x (S-EDA) (respectively, \bar{T}_c^x (ID²C-EDA)) belonging to T_c^{PS-opt} , then $\delta_{dist}^{T_c^x}$ (S-EDA) (respectively, $\delta_{dist}^{T_c^x}$ (ID²C-EDA)) will be less than or equal to 0 (i.e., negative real numbers or 0); otherwise, $\delta_{dist}^{T_c^x}$ (S-EDA) (respectively, $\delta_{dist}^{T_c^x}$ (ID²C-EDA)) will have positive real numbers.

- (3) $\delta_{dist}^{\lambda_x}$ measures the closeness between λ_x^{PS-opt} and coevolved PS-optimizing negotiation strategy $\bar{\lambda}_x^{PS-opt}$ (S-EDA) (respectively, $\bar{\lambda}_x^{PS-opt}$ (ID²C-EDA)) from coevolutionary learning as follows:
 $\delta_{dist}^{\lambda_x}$ (S-EDA) = $\bar{\lambda}_x^{PS-opt}$ (S-EDA) - $\max(\lambda_x^{PS-opt})$ for the coevolution using S-EDA,
 $\delta_{dist}^{\lambda_x}$ (ID²C-EDA) = $\bar{\lambda}_x^{PS-opt}$ (ID²C-EDA) - $\max(\lambda_x^{PS-opt})$ for the coevolution using ID²C-EDA

where we consider the maximum value $\max(\lambda_x^{PS-opt})$ as the basis of closeness because λ_x^{PS-opt} can be represented as a range of strategy. Hence, if S-EDA (respectively, ID²C-EDA) has coevolved $\bar{\lambda}_x^{PS-opt}$ (S-EDA) (respectively, $\bar{\lambda}_x^{PS-opt}$ (ID²C-EDA)) belonging to λ_x^{PS-opt} , then $\delta_{dist}^{\lambda_x}$ (S-EDA) (respectively, $\delta_{dist}^{\lambda_x}$ (ID²C-EDA)) will be less than or equal to 0 (i.e., negative real numbers or 0); otherwise, $\delta_{dist}^{\lambda_x}$ (S-EDA) (respectively, $\delta_{dist}^{\lambda_x}$ (ID²C-EDA)) will have positive real numbers.

Second, for checking whether balanced negotiation outcomes were achieved, we will simply compare the closeness metric of B with that of S for the obtained agreement prices and agreement times, respectively, as follows:

- (1) $\delta_{dist}^{P_c^B}$ (S-EDA) and $\delta_{dist}^{P_c^S}$ (S-EDA) are compared for the coevolution using S-EDA,
 $\delta_{dist}^{P_c^B}$ (ID²C-EDA) and $\delta_{dist}^{P_c^S}$ (ID²C-EDA) are compared for the coevolution using ID²C-EDA

where if $\delta_{dist}^{P_c^B}$ (S-EDA) and $\delta_{dist}^{P_c^S}$ (S-EDA) (respectively, $\delta_{dist}^{P_c^B}$ (ID²C-EDA) and $\delta_{dist}^{P_c^S}$ (ID²C-EDA)) have the same values, then it is determined that S-EDAs (respectively, ID²C-EDAs) achieved balanced agreement prices (for both B and S); otherwise, it is determined that S-EDAs (respectively, ID²C-EDAs) achieved biased agreement prices (for both B and S).

- (2) $\delta_{dist}^{T_c^B}$ (S-EDA) and $\delta_{dist}^{T_c^S}$ (S-EDA) are compared for the coevolution using S-EDA,
 $\delta_{dist}^{T_c^B}$ (ID²C-EDA) and $\delta_{dist}^{T_c^S}$ (ID²C-EDA) are compared for the coevolution using ID²C-EDA

where if $\delta_{dist}^{T_c^B}$ (S-EDA) and $\delta_{dist}^{T_c^S}$ (S-EDA) (respectively, $\delta_{dist}^{T_c^B}$ (ID²C-EDA) and $\delta_{dist}^{T_c^S}$ (ID²C-EDA)) have the same values, then it is determined that S-EDAs (respectively, ID²C-EDAs) achieved balanced agreement times (for both B and S); otherwise, it is determined that S-EDAs (respectively, ID²C-EDAs) achieved biased agreement times (for both B and S).

In addition, for measuring and comparing the coevolution performance between S-EDAs and ID²C-EDAs in terms of the number of generations, we can use N^{Gen} and N^{Re_init} together as a performance measure. This is because the total average number of generations for coevolutionary learning is determined as $(N^{Re_init} \times G^{max}) + N^{Gen}$. Furthermore, N^{Re_init} gives additional information about the coevolution capability of S-EDAs and ID²C-EDAs. For example, there can be the case that \bar{N}^{Re_init} (S-EDA) (respectively, \bar{N}^{Re_init} (ID²C-EDA)) has reached CNT^{max} , which means that S-EDAs (respectively, ID²C-EDAs) does not have enough coevolution capability for achieving converged solutions. The higher value of \bar{N}^{Re_init} (S-EDA) (respectively, \bar{N}^{Re_init} (ID²C-EDA)) indicates insufficient coevolution capability of S-EDAs (respectively, ID²C-EDAs) for achieving converged solutions.

5.2 Observations and analyses

5.2.1 Results of Type-I experiments

For (Long, Mid) in Table 6, (Mid, Short) in Table 7 and (Long, Short) in Table 8, B has a bargaining advantage

Table 6 Results of Type-I experiments in (Long, Mid)

(Long, Mid)		$IP_B = 5, RP_B = 80, \tau_B = 100; IP_S = 95, RP_S = 15, \tau_S = 50$								
Combinations of Weights										
(w_{NP}^B, w_{NS}^B)	(0.7, 0.3)			(0.5, 0.5)						(0.3, 0.7)
dP_c^B	27.5			42.5						57.5
dT_c^B	70			50						30
(w_{NP}^S, w_{NS}^S)	(0.7, 0.3)	(0.5, 0.5)	(0.3, 0.7)	(0.7, 0.3)	(0.5, 0.5)	(0.3, 0.7)	(0.7, 0.3)	(0.5, 0.5)	(0.3, 0.7)	
dP_c^B	71	55	39	71	55	39	71	55	39	
dT_c^B	35	25	15	35	25	15	35	25	15	
PS-Optimization mode	(Mode 1)	(Mode 2)	(Mode 3)	(Mode 4)	(Mode 5)	(Mode 6)	(Mode 7)	(Mode 8)	(Mode 9)	
Agreement Price										
P_c	P_c^{PS-opt}	27.50	27.50	27.50	42.50	42.50	39.00	57.50	55.00	39.00
	\bar{P}_c^B (S-EDA)	27.58 (6.94E-01)	27.67 (7.57E-01)	27.59 (7.45E-01)	42.38 (1.04E+00)	42.25 (1.26E+00)	42.53 (9.89E-01)	55.03 (7.32E+00)	57.32 (8.07E+00)	39.30 (5.70E-02)
	$\delta_{dist}^{P_c^B}$ (S-EDA)	0.08	0.17	0.09	0.12	0.25	3.53	2.47	2.32	0.30
	\bar{P}_c^B (ID ² C-EDA)	27.50 (9.73E-03)	27.55 (2.08E-01)	27.60 (4.65E-01)	42.53 (9.13E-02)	42.59 (4.16E-01)	39.00 (1.54E-03)	57.96 (2.40E-01)	55.25 (7.54E-01)	39.00 (1.07E-03)
	$\delta_{dist}^{P_c^B}$ (ID ² C-EDA)	0	0.05	0.10	0.03	0.09	0	0.46	0.25	0
	\bar{P}_c^S (S-EDA)	59.43 (1.67E+01)	51.99 (1.02E+01)	40.59 (1.31E+01)	66.84 (7.75E+00)	52.96 (8.23E+00)	43.54 (1.58E+01)	65.86 (9.88E+00)	54.60 (4.85E+00)	38.99 (2.55E-02)
	$\delta_{dist}^{P_c^S}$ (S-EDA)	31.93	24.49	13.09	24.34	10.46	4.54	8.36	0.40	0.01
	\bar{P}_c^S (ID ² C-EDA)	27.50 (9.73E-03)	27.55 (2.08E-01)	27.60 (4.65E-01)	42.53 (9.13E-02)	42.59 (4.16E-01)	39.00 (1.54E-03)	57.96 (2.40E-01)	55.25 (7.54E-01)	39.00 (1.07E-03)
	$\delta_{dist}^{P_c^S}$ (ID ² C-EDA)	0	0.05	0.10	0.03	0.09	0	0.46	0.25	0
Agreement Time										
T_c	T_c^{PS-opt}	[1, 35.00]	[1, 25.00]	[1, 15.00]	[1, 35.00]	[1, 25.00]	[1, 15.00]	[1, 35.00]	[1, 25.00]	[1, 15.00]
	\bar{T}_c^B (S-EDA)	45.31 (3.98E+00)	43.90 (4.70E+00)	43.73 (5.17E+00)	39.86 (6.89E+00)	40.58 (6.23E+00)	39.38 (6.48E+00)	27.00 (8.45E+00)	25.95 (8.26E+00)	1.00 (0.00E+00)
	$\delta_{dist}^{T_c^B}$ (S-EDA)	+10.31	+18.9	+28.73	+4.86	+15.58	+24.38	-8.00	+0.95	-14.00
	\bar{T}_c^B (ID ² C-EDA)	35.62 (1.30E+00)	25.14 (2.25E+00)	12.85 (5.12E+00)	35.84 (3.68E-01)	25.51 (6.11E-01)	1.00 (0.00E+00)	35.16 (3.68E-01)	1.13 (4.18E-01)	1.00 (0.00E+00)
	$\delta_{dist}^{T_c^B}$ (ID ² C-EDA)	+0.62	+0.14	-2.15	+0.84	+0.51	-14.00	+0.16	-23.87	-14.00
	\bar{T}_c^S (S-EDA)	29.86 (8.57E+00)	30.05 (1.05E+01)	27.59 (1.39E+01)	29.64 (8.91E+00)	30.35 (1.05E+01)	23.99 (1.36E+01)	30.24 (8.81E+00)	30.80 (9.70E+00)	1.00 (0.00E+00)
	$\delta_{dist}^{T_c^S}$ (S-EDA)	-5.14	+5.05	+12.59	-5.36	+5.35	+8.99	-4.76	+5.80	-14.00
	\bar{T}_c^S (ID ² C-EDA)	35.52 (1.31E+00)	24.90 (2.20E+00)	12.67 (5.03E+00)	35.07 (2.56E-01)	25.08 (4.42E-01)	1.00 (0.00E+00)	35.16 (3.68E-01)	1.13 (4.18E-01)	1.00 (0.00E+00)
	$\delta_{dist}^{T_c^S}$ (ID ² C-EDA)	+0.52	-0.10	-2.33	+0.07	+0.08	-14.00	+0.16	-23.87	-14.00
Strategy										
λ_B	λ_B^{PS-opt}	(0, 1.1468]	(0, 0.8685]	(0, 0.6346]	(0, 0.6603]	(0, 0.5000]	(0, 0.4170]	(0, 0.2962]	(0, 0.2925]	(0, 0.4170]
	$\bar{\lambda}_B^{PS-opt}$ (S-EDA)	1.4905 (1.58E-01)	1.4306 (1.64E-01)	1.4316 (1.85E-01)	0.7502 (1.29E-01)	0.7688 (1.28E-01)	0.7365 (1.26E-01)	0.3175 (1.49E-01)	0.2734 (1.24E-01)	0.0900 (4.10E-02)
	$\delta_{dist}^{\lambda_B}$ (S-EDA)	+0.3437	+0.5621	+0.7970	+0.0899	+0.2688	+0.3195	+0.0213	-0.0191	-0.3270
	$\bar{\lambda}_B^{PS-opt}$ (ID ² C-EDA)	1.1377 (3.83E-02)	0.8517 (5.43E-02)	0.5628 (1.28E-01)	0.6592 (3.32E-03)	0.4980 (1.12E-02)	0.1718 (9.80E-06)	0.3315 (4.33E-03)	0.0856 (6.20E-03)	0.1659 (1.68E-02)
	$\delta_{dist}^{\lambda_B}$ (ID ² C-EDA)	-0.0091	-0.0168	-0.0718	-0.0011	-0.002	-0.2452	+0.0353	-0.2069	-0.2511

Table 6 (Continued)

(Long, Mid)		$IP_B = 5, RP_B = 80, \tau_B = 100; IP_S = 95, RP_S = 15, \tau_S = 50$								
λ_S	λ_S^{PS-opt}	(0, 0.4763]	(0, 0.2451]	(0, 0.1411]	(0, 1.1809]	(0, 0.6077]	(0, 0.2962]	(0, 1.4833]	(0, 1.0000]	(0, 0.2962]
	$\bar{\lambda}_S^{PS-opt}$ (S-EDA)	2.0679 (1.38E+00)	1.7902 (1.31E+00)	1.2374 (1.25E+00)	2.3773 (1.18E+00)	1.9054 (1.37E+00)	1.0098 (1.01E+00)	2.3907 (1.23E+00)	1.8967 (1.23E+00)	0.0911 (1.17E-04)
	$\delta_{dist}^{\lambda_B}$ (S-EDA)	+1.5916	+1.5451	+1.0963	+1.1964	+1.2977	+0.7136	+0.9074	+0.8967	-0.2051
	$\bar{\lambda}_S^{PS-opt}$ (ID ² C-EDA)	0.4778 (3.97E-02)	0.2423 (2.56E-02)	0.1252 (3.46E-02)	1.1825 (4.92E-03)	0.6101 (1.13E-02)	0.0912 (7.00E-06)	2.1688 (1.68E-02)	0.1830 (1.76E-02)	0.0912 (4.87E-06)
	$\delta_{dist}^{\lambda_S}$ (ID ² C-EDA)	+0.0015	-0.0028	-0.0159	+0.0016	+0.0024	-0.2050	+0.6855	-0.8170	-0.2050
Convergence of EDA										
N^{Gen}	\bar{N}^{Gen} (S-EDA)	2500	2500	2500	2500	2500	2500	2500	2500	33.25
	\bar{N}^{Gen} (ID ² C-EDA)	67.87	144.23	316.66	95.72	103.40	380.22	666.10	983.24	44.42
N^{Re_init}	\bar{N}^{Re_init} (S-EDA)	10	10	10	10	10	10	10	10	0
	\bar{N}^{Re_init} (ID ² C-EDA)	0	0	0.041	0	0	0.014	0.012	0.521	0

Table 7 Results of Type-I experiments in (Mid, Short)

(Mid, Short)		$IP_B = 5, RP_B = 80, \tau_B = 50; IP_S = 95, RP_S = 15, \tau_S = 25$								
Combinations of Weights										
(w_{NP}^B, w_{NS}^B)	(0.7, 0.3)			(0.5, 0.5)			(0.3, 0.7)			
dP_c^B	27.5			42.5			57.5			
dT_c^B	35			25			15			
(w_{NP}^S, w_{NS}^S)	(0.7, 0.3)	(0.5, 0.5)	(0.3, 0.7)	(0.7, 0.3)	(0.5, 0.5)	(0.3, 0.7)	(0.7, 0.3)	(0.5, 0.5)	(0.3, 0.7)	
dP_c^B	71	55	39	71	55	39	71	55	39	
dT_c^B	14	10	6	14	10	6	14	10	6	
PS-Optimization mode	(Mode 1)	(Mode 2)	(Mode 3)	(Mode 4)	(Mode 5)	(Mode 6)	(Mode 7)	(Mode 8)	(Mode 9)	
Agreement Price										
P_c	P_c^{PS-opt}	27.50	27.50	27.50	42.50	42.50	39.00	57.5	55.00	39.00
	\bar{P}_c^B (S-EDA)	27.45 (9.81E-01)	27.49 (9.65E-01)	27.40 (1.01E+00)	42.67 (1.90E+00)	42.20 (1.91E+00)	42.12 (2.54E+00)	56.80 (4.94E+00)	56.71 (4.47E+00)	39.31 (6.51E-02)
$\delta_{dist}^{P_c^B}$	(S-EDA)	0.05	0.01	0.10	0.17	0.30	3.12	0.70	1.71	0.31
	(ID ² C-EDA)	27.50 (1.12E-02)	27.57 (2.64E-01)	27.54 (6.19E-01)	42.79 (4.34E-01)	42.66 (4.94E-01)	39.00 (1.37E-02)	58.39 (2.64E-01)	55.30 (8.16E-01)	39.00 (6.12E-04)
$\delta_{dist}^{P_c^B}$	(ID ² C-EDA)	0	0.07	0.04	0.29	0.16	0	0.89	0.30	0
	\bar{P}_c^S (S-EDA)	64.21 (1.19E+01)	53.81 (6.75E+00)	46.03 (1.39E+01)	66.55 (8.82E+00)	54.31 (6.24E+00)	43.51 (1.32E+01)	65.40 (9.78E+00)	53.89 (7.40E+00)	39.00 (2.70E-02)
$\delta_{dist}^{P_c^S}$	(S-EDA)	36.71	26.31	18.53	24.05	11.81	4.51	7.9	1.11	0.00
	(ID ² C-EDA)	27.50 (1.12E-02)	27.57 (2.64E-01)	27.54 (6.19E-01)	42.79 (4.34E-01)	42.66 (4.94E-01)	39.00 (1.37E-02)	58.39 (2.64E-01)	55.30 (8.16E-01)	39.00 (6.12E-04)
$\delta_{dist}^{P_c^S}$	(ID ² C-EDA)	0	0.07	0.04	0.29	0.16	0	0.89	0.30	0
	Agreement Time									
T_c	T_c^{PS-opt}	[1, 14.00]	[1, 10.00]	[1, 6.00]	[1, 14.00]	[1, 10.00]	[1, 6.00]	[1, 14.00]	[1, 10.00]	[1, 6.00]
	\bar{T}_c^B (S-EDA)	18.04 (2.87E+00)	18.25 (1.77E+00)	18.30 (2.26E+00)	15.55 (3.41E+00)	15.98 (3.18E+00)	13.60 (5.99E+00)	13.36 (2.98E+00)	12.81 (3.45E+00)	1.00 (0.00E+00)
	$\delta_{dist}^{T_c^B}$ (S-EDA)	+4.04	+8.25	+12.30	+1.55	+5.98	+7.60	-0.64	+2.81	-5.00

Table 7 (Continued)

(Mid, Short)		$IP_B = 5, RP_B = 80, \tau_B = 50; IP_S = 95, RP_S = 15, \tau_S = 25$								
\bar{T}_c^B (ID ² C-EDA)		14.69 (5.26E-01)	10.31 (1.39E+00)	5.84 (1.80E+00)	14.64 (4.82E-01)	10.46 (5.01E-01)	1.00 (0.00E+00)	14.65 (4.79E-01)	1.13 (3.38E-01)	1.00 (0.00E+00)
$\delta_{dist}^{T_c^B}$ (ID ² C-EDA)		+0.69	+0.31	-0.16	+0.64	+0.46	-5.00	+0.65	-8.87	-5.00
\bar{T}_c^S (S-EDA)		11.87 (3.15E+00)	12.49 (3.82E+00)	9.29 (5.57E+00)	12.39 (3.52E+00)	12.41 (3.79E+00)	8.47 (5.74E+00)	11.93 (3.41E+00)	11.96 (3.74E+00)	1.00 (0.00E+00)
$\delta_{dist}^{T_c^S}$ (S-EDA)		-2.13	+2.49	+3.29	-1.61	+2.41	+2.47	-2.07	+1.96	-5.00
\bar{T}_c^S (ID ² C-EDA)		14.69 (5.44E-01)	10.03 (1.32E+00)	5.64 (1.70E+00)	14.25 (4.35E-01)	10.06 (2.39E-01)	1.00 (0.00E+00)	14.14 (3.49E-01)	1.13 (3.38E-01)	1.00 (0.00E+00)
$\delta_{dist}^{T_c^S}$ (ID ² C-EDA)		+0.69	+0.03	-0.36	+0.25	+0.06	-5.00	+0.14	-8.87	-5.00
Strategy										
λ_B	λ_B^{PS-opt}	(0, 0.9458]	(0, 0.7481]	(0, 0.5678]	(0, 0.5445]	(0, 0.4307]	(0, 0.3731]	(0, 0.2802]	(0, 0.2519]	(0, 0.3731]
	$\bar{\lambda}_B^{PS-opt}$ (S-EDA)	1.1341 (1.59E-01)	1.1421 (1.17E-01)	1.1540 (1.34E-01)	0.5690 (1.12E-01)	0.5935 (1.14E-01)	0.5037 (2.04E-01)	0.2740 (8.90E-02)	0.2635 (7.88E-02)	0.1015 (5.13E-02)
	$\delta_{dist}^{\lambda_B}$ (S-EDA)	+0.1883	+0.3940	+0.5862	+0.0245	+0.1628	+0.1306	-0.0062	+0.0116	-0.2716
	$\bar{\lambda}_B^{PS-opt}$ (ID ² C-EDA)	0.9343 (2.56E-02)	0.7221 (6.41E-02)	0.5259 (9.23E-02)	0.5380 (9.53E-03)	0.4275 (8.65E-03)	0.2022 (1.03E-04)	0.2670 (3.90E-03)	0.1008 (5.35E-03)	0.1993 (1.20E-02)
	$\delta_{dist}^{\lambda_B}$ (ID ² C-EDA)	-0.0115	-0.0260	-0.0419	-0.0065	-0.0032	-0.1709	-0.0132	-0.1511	-0.1738
λ_S	λ_S^{PS-opt}	(0, 0.4763]	(0, 0.2451]	(0, 0.1411]	(0, 1.1809]	(0, 0.6077]	(0, 0.2962]	(0, 2.1243]	(0, 1.0000]	(0, 0.2962]
	$\bar{\lambda}_S^{PS-opt}$ (S-EDA)	2.1689 (1.27E+00)	1.8539 (1.21E+00)	1.1001 (1.21E+00)	2.5325 (1.28E+00)	1.8229 (1.21E+00)	0.8535 (8.94E-01)	2.2375 (1.25E+00)	1.6268 (1.07E+00)	0.1190 (1.61E-04)
	$\delta_{dist}^{\lambda_S}$ (S-EDA)	+1.6926	+1.6088	+0.9590	+1.3516	+1.2152	+0.5573	+0.1132	+0.6268	-0.1772
	$\bar{\lambda}_S^{PS-opt}$ (ID ² C-EDA)	0.4824 (3.44E-02)	0.2403 (3.63E-02)	0.1328 (3.16E-02)	1.1973 (2.46E-02)	0.6128 (1.51E-02)	0.1191 (8.16E-05)	2.1941 (1.71E-02)	0.2424 (2.91E-02)	0.1191 (3.66E-06)
	$\delta_{dist}^{\lambda_S}$ (ID ² C-EDA)	+0.0061	-0.0048	-0.0083	+0.0164	+0.0051	-0.1771	+0.0698	-0.7576	-0.1771
Convergence of EDA										
N^{Gen}	\bar{N}^{Gen} (S-EDA)	2500	2500	2500	2500	2500	2315.76	2500	2500	30.72
	\bar{N}^{Gen} (ID ² C-EDA)	63.85	95.64	133.86	96.58	98.16	889.92	78.02	414.8	43.99
N^{Re_init}	\bar{N}^{Re_init} (S-EDA)	10	10	10	10	10	9.317	10	10	0
	\bar{N}^{Re_init} (ID ² C-EDA)	0	0	0	0	0	0.643	0	0.112	0

Table 8 Results of Type-I experiments in (Long, Short)

(Long, Short)		$IP_B = 5, RP_B = 80, \tau_B = 100; IP_S = 95, RP_S = 15, \tau_S = 25$								
Combinations of Weights										
(w_{NP}^B, w_{NS}^B)		(0.7, 0.3)			(0.5, 0.5)			(0.3, 0.7)		
dP_c^B		27.5			42.5			39		
dT_c^B		70			50			6		
(w_{NP}^S, w_{NS}^S)		(0.7, 0.3)	(0.5, 0.5)	(0.3, 0.7)	(0.7, 0.3)	(0.5, 0.5)	(0.3, 0.7)	(0.7, 0.3)	(0.5, 0.5)	(0.3, 0.7)
dP_c^S		71	55	39	71	55	39	71	55	39
dT_c^S		14	10	6	14	10	6	14	10	6
PS-Optimization mode		(Mode 1)	(Mode 2)	(Mode 3)	(Mode 4)	(Mode 5)	(Mode 6)	(Mode 7)	(Mode 8)	(Mode 9)
Agreement Price										
P_c	P_c^{PS-opt}	27.50	27.50	27.50	42.50	42.50	39.00	57.50	55.00	39.00

Table 8 (Continued)

(Long, Short)		$IP_B = 5, RP_B = 80, \tau_B = 100; IP_S = 95, RP_S = 15, \tau_S = 25$								
\bar{P}_c^B (S-EDA)		27.42 (2.61E+00)	27.48 (2.65E+00)	27.18 (2.67E+00)	41.57 (3.81E+00)	42.40 (3.81E+00)	39.78 (2.13E+00)	57.13 (2.75E+00)	54.98 (7.32E+00)	39.30 (5.38E-02)
$\delta_{dist}^{P_c^B}$ (S-EDA)		0.08	0.02	0.32	0.93	0.10	0.78	0.37	0.02	0.30
\bar{P}_c^B (ID ² C-EDA)		27.50 (2.87E-03)	27.68 (4.18E-01)	27.80 (8.90E-01)	43.30 (2.12E-01)	42.56 (2.81E+00)	39.00 (2.25E-05)	57.72 (4.32E+00)	55.15 (5.97E-01)	39.00 (1.00E-05)
$\delta_{dist}^{P_c^B}$ (ID ² C-EDA)		0	0.18	0.30	0.80	0.06	0	0.22	0.15	0
\bar{P}_c^S (S-EDA)		60.12 (1.61E+01)	51.42 (1.34E+01)	46.54 (1.48E+01)	59.12 (1.58E+01)	51.36 (1.58E+01)	39.36 (5.74E+00)	58.43 (6.22E+00)	50.78 (1.31E+01)	39.00 (3.06E-02)
$\delta_{dist}^{P_c^S}$ (S-EDA)		32.62	23.92	19.04	16.62	8.86	0.36	0.93	4.22	0
\bar{P}_c^S (ID ² C-EDA)		27.50 (2.87E-03)	27.68 (4.18E-01)	27.80 (8.90E-01)	43.30 (2.12E-01)	42.56 (2.81E+00)	39.00 (2.00E-05)	57.72 (4.32E+00)	55.15 (5.97E-01)	39.00 (1.00E-05)
$\delta_{dist}^{P_c^S}$ (ID ² C-EDA)		0	0.18	0.30	0.80	0.06	0	0.22	0.15	0
Agreement Time										
T_c	T_c^{PS-opt}	[1, 14.00]	[1, 10.00]	[1, 6.00]	[1, 14.00]	[1, 10.00]	[1, 6.00]	[1, 14.00]	[1, 10.00]	[1, 6.00]
	\bar{T}_c^B (S-EDA)	17.79 (3.56E+00)	18.05 (2.97E+00)	18.16 (3.12E+00)	15.79 (3.30E+00)	15.84 (3.30E+00)	3.73 (6.04E+00)	13.96 (1.46E+00)	14.59 (3.82E+00)	1.00 (0.00E+00)
	$\delta_{dist}^{T_c^B}$ (S-EDA)	+3.79	+8.05	+12.16	+1.79	+5.84	-2.27	-0.04	+4.59	-5.00
	\bar{T}_c^B (ID ² C-EDA)	14.81 (7.06E-01)	10.56 (7.01E-01)	5.57 (1.97E+00)	14.64 (4.82E-01)	10.35 (1.07E+00)	1.00 (0.00E+00)	14.45 (7.44E-01)	1.07 (2.56E-01)	1.00 (0.00E+00)
	$\delta_{dist}^{T_c^B}$ (ID ² C-EDA)	+0.81	+0.56	-0.43	+0.64	+0.35	-5.00	+0.45	-8.93	-5.00
	\bar{T}_c^S (S-EDA)	12.03 (3.80E+00)	11.58 (4.19E+00)	11.32 (5.57E+00)	12.26 (3.71E+00)	11.29 (3.71E+00)	2.90 (4.75E+00)	13.77 (1.26E+00)	11.93 (4.91E+00)	1.01 (1.00E-01)
	$\delta_{dist}^{T_c^S}$ (S-EDA)	-1.97	+1.58	+5.32	-1.74	+1.29	-3.10	-0.23	+1.93	-4.99
	\bar{T}_c^S (ID ² C-EDA)	14.76 (7.26E-01)	10.33 (6.82E-01)	5.36 (1.86E+00)	14.27 (4.46E-01)	10.10 (1.00E+00)	1.00 (0.00E+00)	14.16 (6.62E-01)	1.07 (2.56E-01)	1.00 (0.00E+00)
	$\delta_{dist}^{T_c^S}$ (ID ² C-EDA)	+0.76	+0.33	-0.64	+0.27	+0.10	-5.00	+0.16	-8.93	-5.00
Strategy										
λ_B	λ_B^{PS-opt}	(0, 0.6124]	(0, 0.5229]	(0, 0.4279]	(0, 0.3525]	(0, 0.3010]	(0, 0.2812]	(0, 0.1814]	(0, 0.1761]	(0, 0.2812]
	$\bar{\lambda}_B^{PS-opt}$ (S-EDA)	0.6829 (1.21E-01)	0.6864 (1.04E-01)	0.6968 (1.24E-01)	0.3816 (8.48E-02)	0.3699 (8.48E-02)	0.1322 (1.15E-01)	0.1796 (3.33E-02)	0.2122 (9.93E-02)	0.0883 (3.89E-02)
	$\delta_{dist}^{\lambda_B}$ (S-EDA)	+0.0705	+0.1635	+0.2689	+0.0291	+0.0689	-0.1490	-0.0018	+0.0361	-0.1929
	$\bar{\lambda}_B^{PS-opt}$ (ID ² C-EDA)	0.6102 (1.48E-02)	0.5161 (1.71E-02)	0.3898 (6.17E-02)	0.3418 (2.84E-03)	0.3441 (4.70E-01)	0.1716 (2.13E-03)	0.1858 (1.08E-01)	0.0870 (3.22E-03)	0.1703 (8.38E-03)
	$\delta_{dist}^{\lambda_B}$ (ID ² C-EDA)	-0.0022	-0.0068	-0.0381	-0.0107	+0.0431	-0.1096	+0.0044	-0.0891	-0.1109
λ_S	λ_S^{PS-opt}	(0, 0.4763]	(0, 0.2451]	(0, 0.1411]	(0, 1.1809]	(0, 0.6077]	(0, 0.2962]	(0, 2.1243]	(0, 1.0000]	(0, 0.2962]
	$\bar{\lambda}_S^{PS-opt}$ (S-EDA)	2.0434 (1.39E+00)	1.5194 (1.21E+00)	1.3860 (1.17E+00)	2.0242 (1.30E+00)	1.4509 (1.30E+00)	0.3333 (6.62E-01)	2.1387 (5.13E-01)	1.7600 (1.37E+00)	0.1190 (1.89E-04)
	$\delta_{dist}^{\lambda_S}$ (S-EDA)	+1.5671	+1.2743	+1.2449	+0.8433	+0.8432	+0.0371	+0.0144	+0.7600	-0.1772
	$\bar{\lambda}_S^{PS-opt}$ (ID ² C-EDA)	0.5103 (2.86E-01)	0.2474 (1.90E-02)	0.1313 (3.42E-02)	1.2241 (1.15E-02)	0.6612 (4.38E-01)	0.1191 (1.00E-07)	2.2020 (2.83E-01)	0.2366 (1.91E-02)	0.1191 (1.00E-07)
	$\delta_{dist}^{\lambda_S}$ (ID ² C-EDA)	+0.0340	+0.0023	-0.0098	+0.0432	+0.0535	-0.1771	+0.0777	-0.7634	-0.1771
Convergence of EDA										
N^{Gen}	\bar{N}^{Gen} (S-EDA)	2500	2500	2500	2500	2500	1502.13	1528.36	2500	32.89
	\bar{N}^{Gen} (ID ² C-EDA)	55.55	91.5	96.33	81.38	90.25	1062.42	72.61	309.81	44.79
N^{Re_init}	\bar{N}^{Re_init} (S-EDA)	10	10	10	10	10	4.374	4.590	10	0
	\bar{N}^{Re_init} (ID ² C-EDA)	0	0	0	0	0	0.382	0	0.243	0

over S in terms of time. Furthermore, in (Long, Mid), (Mid, Short) and (Long, Short), dP_c^B and $\min(dT_c^B, dT_c^S)$ —which determine the optimum agreement points of all optimization modes 1 to 9—are in the range of NSS. Hence, we calculated both P_c^{PS-opt} and T_c^{PS-opt} from Theorem 3, λ_B^{PS-opt} from (12) and λ_S^{PS-opt} from (13). From the results in Tables 6 to 8, the following two observations were drawn.

Observation 1 While S-EDAs generally could not obtain effective PS-optimizing negotiation outcomes and coevolve effective PS-optimizing negotiation strategies in most of the optimization modes, ID²C-EDAs generally obtained effective PS-optimizing negotiation outcomes and coevolved effective PS-optimizing negotiation strategies in all the optimization modes.

Analysis It can be observed from Tables 6 to 8 that: (1) the values of $\delta_{dist}^{P^B}$ (S-EDA) and $\delta_{dist}^{P^S}$ (S-EDA) were large (especially, for S), and the difference between $\delta_{dist}^{P^B}$ (S-EDA) and $\delta_{dist}^{P^S}$ (S-EDA) was too large to be effective agreement prices in most of the optimization modes, (2) the values of $\delta_{dist}^{T^B}$ (S-EDA) and $\delta_{dist}^{T^S}$ (S-EDA) were large positive real numbers and the difference between $\delta_{dist}^{T^B}$ (S-EDA) and $\delta_{dist}^{T^S}$ (S-EDA) was too large to be effective agreement times in most of the optimization modes, and (3) the values of $\delta_{dist}^{\lambda^B}$ (S-EDA) and $\delta_{dist}^{\lambda^S}$ (S-EDA) were large positive real numbers (especially, for S) to be effective PS-optimizing negotiation strategies in most of the optimization modes. These indicate that S-EDAs generally: (1) achieved both ineffective agreement prices (especially, for S) and agreement times and (2) coevolved ineffective PS-optimizing negotiation strategies in terms of both closeness to the optimum and balanced coevolution. This was mainly because of different coevolution speed between POP_B and POP_S ; since POP_B has converged to around an optimal value more rapidly than POP_S , POP_S could not have sufficient diversity (of opponents) to optimize its solutions in making coevolutionary interactions and hence, S-EDAs could not achieve effective and balanced solutions. This phenomenon can be considered as the premature convergence in the coevolution situation.

It can be also observed from Tables 6 to 8 that: (1) both the values of $\delta_{dist}^{P^B}$ (ID²C-EDA) and $\delta_{dist}^{P^S}$ (ID²C-EDA) were either small or optimal, and the difference between $\delta_{dist}^{P^B}$ (ID²C-EDA) and $\delta_{dist}^{P^S}$ (ID²C-EDA) was also small in all the optimization modes; (2) the values of $\delta_{dist}^{T^B}$ (ID²C-EDA) and $\delta_{dist}^{T^S}$ (ID²C-EDA) were small or in the range of the optimum and the difference between $\delta_{dist}^{T^B}$ (ID²C-EDA) and $\delta_{dist}^{T^S}$ (ID²C-EDA) was also small in most of the optimization modes; and (3) the values of $\delta_{dist}^{\lambda^B}$ (ID²C-EDA) and

$\delta_{dist}^{\lambda^S}$ (ID²C-EDA) were small in most of the optimization modes. These indicate that ID²C-EDAs generally could: (1) achieve both effective agreement prices and agreement times and (2) coevolve effective PS-optimizing negotiation strategies in terms of both closeness to the optimum and balanced coevolution. This was because ID²C-EDAs have sufficient capability for achieving close to optimal and balanced solutions for both B and S by dynamically adjusting the degree of intensification and diversification of POP_B and POP_S using DR. Furthermore, by adopting LNS and PR, it is possible for ID²C-EDAs to improve solution accuracy and to avoid inappropriate population configurations, respectively.

Observation 2 While S-EDAs could not coevolve converged populations, ID²C-EDAs generally coevolved converged populations.

Analysis It can be observed from Tables 6 to 8 that the values of \bar{N}^{Re_init} (S-EDA) reached CNT^{\max} in most of the optimization modes while the values of \bar{N}^{Re_init} (ID²C-EDA) were much smaller than \bar{N}^{Re_init} (S-EDA) (and were often close to 0) in all the optimization modes. Hence, comparing to ID²C-EDAs, S-EDAs required an extremely larger number of (total) generations for the coevolution. This indicates that S-EDAs generally did not have enough capability for coevolving converged populations while ID²C-EDAs had enough capability for coevolving converged populations within a reasonable number of generations. The reason that ID²C-EDAs outperformed S-EDAs is mainly due to the innovation of DR (in the coevolution process) which allows ID²C-EDAs to search for promising solutions adaptively. In DR, the diversification procedure helps to avoid premature convergence in a population by maintaining population diversity to a certain level and the refinement procedure helps to achieve optimal solutions by generating promising solutions using regional population history information and replacing less feasible solutions with the generated promising solutions. Furthermore, LNS can contribute to resolving the problem of the late convergence of populations, and PR assists to avoid configuring inappropriate populations in early generations. Hence, by adopting DR together with LNS and PR, each ID²C-EDA is more likely to escape premature convergence and maintain enough population diversity, which enables ID²C-EDAs to coevolve converged populations in the coevolutionary learning.

From these Observations 1 and 2, we can draw the following conclusion for the Type-I experiments.

Conclusion 1 When the negotiation setting with sufficiently large AgZ^{NP} is provided for PS-optimizing agents, ID²C-EDAs generally coevolve effective (converged) PS-optimizing negotiation strategies for both B and S while

S-EDA generally fails to coevolve such negotiation strategies within reasonable numbers of generations in most of the cases.

5.2.2 Results of Type-II experiments

For (Long, Mid) in Table 9, (Mid, Short) in Table 10 and (Long, Short) in Table 11, B has a bargaining advantage over S in terms of time. In addition, in (Long, Mid), (Mid, Short) and (Long, Short), dP_c^B and $\min(dT_c^B, dT_c^S)$ of the optimization modes 1 to 6 are not in the range of NSS. Hence, we calculated both P_c^{PS-opt} and T_c^{PS-opt} from Theorem 5, λ_B^{PS-opt} from (12) and λ_S^{PS-opt} from (13). In contrast, in (Long, Mid), (Mid, Short) and (Long, Short), dP_c^B and $\min(dT_c^B, dT_c^S)$ of the optimization modes 7 to 9 are in the range of NSS. Hence, we calculated both P_c^{PS-opt} and T_c^{PS-opt} from Theorem 3, λ_B^{PS-opt} from (12) and λ_S^{PS-opt} from (13). From the results in Tables 9 to 11, the following three observations were drawn.

Observation 3 In optimization modes 1 to 6, both S-EDAs and ID²C-EDAs generally achieved effective PS-optimizing negotiation outcomes and coevolved effective PS-optimizing negotiation strategies.

Analysis From the results of the optimization modes 1 to 6 in Tables 9 to 11, it can be seen that: (1) the values of $\delta_{dist}^{P_c^B}$ (S-EDA) and $\delta_{dist}^{P_c^S}$ (S-EDA) were small and the difference between $\delta_{dist}^{P_c^B}$ (S-EDA) and $\delta_{dist}^{P_c^S}$ (S-EDA) was also small, (2) the values of $\delta_{dist}^{T_c^B}$ (S-EDA) and $\delta_{dist}^{T_c^S}$ (S-EDA) were in the range of the optimum and the difference between $\delta_{dist}^{T_c^B}$ (S-EDA) and $\delta_{dist}^{T_c^S}$ (S-EDA) was small, and (3) the values of $\delta_{dist}^{\lambda_B}$ (S-EDA) and $\delta_{dist}^{\lambda_S}$ (S-EDA) were in the range of the optimum. These indicate that S-EDAs generally could: (1) obtain effective agreement prices and agreement times and (2) coevolve effective PS-optimizing negotiation strategies in terms of both closeness to optimum and balanced coevolution. Similarly, from all the results of the optimization modes 1 to 6 in Tables 9 to 11, it can be seen that: (1) the values of $\delta_{dist}^{P_c^B}$ (ID²C-EDA) and $\delta_{dist}^{P_c^S}$ (ID²C-EDA) were 0 and the difference between $\delta_{dist}^{P_c^B}$ (ID²C-EDA) and $\delta_{dist}^{P_c^S}$ (ID²C-EDA) was also 0; (2) the values of $\delta_{dist}^{T_c^B}$ (ID²C-EDA) and $\delta_{dist}^{T_c^S}$ (ID²C-EDA) were 0, and the difference between $\delta_{dist}^{T_c^B}$ (ID²C-EDA) and $\delta_{dist}^{T_c^S}$ (ID²C-EDA) was also 0; and (3) the values of $\delta_{dist}^{\lambda_B}$ (ID²C-EDA) and $\delta_{dist}^{\lambda_S}$ (ID²C-EDA) were in the range of the optimum. In summary, both S-EDAs and ID²C-EDAs could obtain optimal PS-optimizing negotiation outcomes and coevolve optimal PS-optimizing negotiation strategies for these modes. In the results, the coevolved negotiation agreements were made at

RP_S and τ_S . This is because B adopting the time-dependent negotiation strategy achieves all of its payoff at τ_S by accepting S 's final proposal RP_S . Since we designed the fitness function putting more emphasis on optimizing price than optimizing speed by setting $\rho_{NP} = w_{NP}^x/100$ and $\rho_{NS} = w_{NS}^x$, S-EDAs and ID²C-EDAs are less likely to make price concessions for achieving rapid agreements and strictly hold RP_S as the optimal agreement price. This is the reason of the successful coevolution performance of S-EDAs for the above cases (when it is compared to the Type-I experiments).

Observation 4 In the optimization modes 7 to 9, ID²C-EDAs generally obtained effective PS-optimizing negotiation outcomes and coevolved effective PS-optimizing negotiation strategies while S-EDAs did not.

Analysis From the results of the optimization modes 7 to 9 in Tables 9 to 11, it can be seen that: (1) the values of $\delta_{dist}^{P_c^B}$ (S-EDA) and $\delta_{dist}^{P_c^S}$ (S-EDA) were large and the difference between $\delta_{dist}^{P_c^B}$ (S-EDA) and $\delta_{dist}^{P_c^S}$ (S-EDA) was too large to be effective agreement prices; (2) the values of $\delta_{dist}^{T_c^B}$ (S-EDA) and $\delta_{dist}^{T_c^S}$ (S-EDA) were large positive real numbers and the difference between $\delta_{dist}^{T_c^B}$ (S-EDA) and $\delta_{dist}^{T_c^S}$ (S-EDA) was too large to be effective agreement times; and (3) the values of $\delta_{dist}^{\lambda_B}$ (S-EDA) and $\delta_{dist}^{\lambda_S}$ (S-EDA) were large positive real numbers to be effective PS-optimizing negotiation strategies. These indicate that S-EDAs generally could: (1) obtain both ineffective agreement prices and agreement times and (2) coevolve ineffective PS-optimizing negotiation strategies in terms of both closeness to optimum and balanced coevolution. In contrast, from all the results of the optimization modes 7 to 9 in Tables 9 to 11, it can also be observed that: (1) both the values of $\delta_{dist}^{P_c^B}$ (ID²C-EDA) and $\delta_{dist}^{P_c^S}$ (ID²C-EDA) were small and the difference between $\delta_{dist}^{P_c^B}$ (ID²C-EDA) and $\delta_{dist}^{P_c^S}$ (ID²C-EDA) is also small; (2) the values of $\delta_{dist}^{T_c^B}$ (ID²C-EDA) and $\delta_{dist}^{T_c^S}$ (ID²C-EDA) were large positive real numbers and the difference between $\delta_{dist}^{T_c^B}$ (ID²C-EDA) and $\delta_{dist}^{T_c^S}$ (ID²C-EDA) was small; and (3) the values of $\delta_{dist}^{\lambda_B}$ (ID²C-EDA) and $\delta_{dist}^{\lambda_S}$ (ID²C-EDA) were small. These indicate that ID²C-EDAs generally could: (1) achieve both effective agreement prices and agreement times and (2) coevolve effective PS-optimizing negotiation strategies in terms of both closeness to optimum and balanced coevolution.

Observation 5 S-EDAs could not coevolve converged populations in some cases while ID²C-EDAs generally coevolved converged populations in most of the cases.

Table 9 Results of Type-II experiments in (Long, Mid)

(Long, Mid)		$IP_B = 15, RP_B = 60, \tau_B = 100; IP_S = 85, RP_S = 40, \tau_S = 50$								
Combinations of Weights										
(w_{NP}^B, w_{NS}^B)	(0.7, 0.3)		(0.5, 0.5)		(0.3, 0.7)					
dP_c^B	28.5		37.5		46.5					
dT_c^B	70		50		30					
(w_{NP}^S, w_{NS}^S)	(0.7, 0.3)	(0.5, 0.5)	(0.3, 0.7)	(0.7, 0.3)	(0.5, 0.5)	(0.3, 0.7)	(0.7, 0.3)	(0.5, 0.5)	(0.3, 0.7)	
dP_c^B	71.5	62.5	53.5	71.5	62.5	53.5	71.5	62.5	53.5	
dT_c^B	35	25	15	35	25	15	35	25	15	
PS-Optimization mode	(Mode 1)	(Mode 2)	(Mode 3)	(Mode 4)	(Mode 5)	(Mode 6)	(Mode 7)	(Mode 8)	(Mode 9)	
Agreement Price										
P_c	P_c^{PS-opt}	40.00	40.00	40.00	40.00	40.00	40.00	46.50	46.50	46.50
	\bar{P}_c^B (S-EDA)	40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	52.09 (1.87E+00)	47.32 (4.43E+00)	45.52 (1.65E+00)
	$\delta_{dist}^{P_c^B}$ (S-EDA)	0	0	0	0	0	0	5.59	0.82	0.98
	\bar{P}_c^B (ID ² C-EDA)	40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	47.13 (8.22E-01)	46.51 (5.23E-02)	46.54 (1.07E-03)
	$\delta_{dist}^{P_c^B}$ (ID ² C-EDA)	0	0	0	0	0	0	0.63	0.01	0.04
	\bar{P}_c^S (S-EDA)	40.07 (1.61E-01)	40.08 (1.51E-01)	40.52 (3.35E-01)	40.05 (1.09E-01)	40.06 (1.16E-01)	40.39 (1.97E-01)	59.50 (4.25E-01)	51.76 (7.14E+00)	52.12 (5.80E+00)
	$\delta_{dist}^{P_c^S}$ (S-EDA)	0.07	0.08	0.52	0.05	0.06	0.39	13.00	5.26	5.62
	\bar{P}_c^S (ID ² C-EDA)	40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	47.13 (8.22E-01)	46.51 (5.23E-02)	46.54 (1.07E-03)
	$\delta_{dist}^{P_c^S}$ (ID ² C-EDA)	0	0	0	0	0	0	0.63	0.01	0.04
Agreement Time										
T_c	T_c^{PS-opt}	[1, 50.00]	[1, 50.00]	[1, 50.00]	[1, 50.00]	[1, 50.00]	[1, 50.00]	[1, 35.00]	[1, 25.00]	[1, 15.00]
	\bar{T}_c^B (S-EDA)	50.00 (0.00E+00)	50.00 (0.00E+00)	50.00 (0.00E+00)	50.00 (0.00E+00)	50.00 (0.00E+00)	50.00 (0.00E+00)	17.84 (8.93E+00)	41.50 (9.83E+00)	44.85 (4.83E+00)
	$\delta_{dist}^{T_c^B}$ (S-EDA)	0	0	0	0	0	0	-17.16	+16.50	+29.85
	\bar{T}_c^B (ID ² C-EDA)	50.00 (0.00E+00)	50.00 (0.00E+00)	50.00 (0.00E+00)	50.00 (0.00E+00)	50.00 (0.00E+00)	50.00 (0.00E+00)	39.09 (2.28E+00)	25.83 (5.79E-01)	15.73 (0.00E+00)
	$\delta_{dist}^{T_c^B}$ (ID ² C-EDA)	0	0	0	0	0	0	+4.09	+0.83	+0.73
	\bar{T}_c^S (S-EDA)	50.00 (0.00E+00)	50.00 (0.00E+00)	50.00 (3.16E-02)	50.00 (0.00E+00)	50.00 (0.00E+00)	50.00 (0.00E+00)	26.92 (6.94E+00)	37.30 (7.94E+00)	42.62 (3.48E+00)
	$\delta_{dist}^{T_c^S}$ (S-EDA)	0	0	0	0	0	0	-8.08	+12.30	+27.62
	\bar{T}_c^S (ID ² C-EDA)	50.00 (0.00E+00)	50.00 (0.00E+00)	50.00 (0.00E+00)	50.00 (0.00E+00)	50.00 (0.00E+00)	50.00 (0.00E+00)	35.16 (2.28E+00)	25.22 (6.02E-01)	15.21 (0.00E+00)
	$\delta_{dist}^{T_c^S}$ (ID ² C-EDA)	0	0	0	0	0	0	+0.16	+0.22	+0.21
Strategy										
λ_B	λ_B^{PS-opt}	(0, 0.8480]	(0, 0.8480]	(0, 0.8480]	(0, 0.8480]	(0, 0.8480]	(0, 0.8480]	(0, 0.3397]	(0, 0.2573]	(0, 0.1880]
	$\bar{\lambda}_B^{PS-opt}$ (S-EDA)	0.8358 (1.58E-01)	0.8356 (6.82E-03)	0.8359 (6.68E-03)	0.8355 (6.72E-03)	0.8354 (6.77E-03)	0.8356 (6.82E-03)	0.1048 (3.34E-02)	0.3967 (1.57E-01)	0.4776 (8.15E-02)
	$\delta_{dist}^{\lambda_B}$ (S-EDA)	-0.0122	-0.0124	-0.0121	-0.0125	-0.0126	-0.0124	-0.2349	+0.1394	+0.2896
	$\bar{\lambda}_B^{PS-opt}$ (ID ² C-EDA)	0.8480 (6.68E-03)	0.8480 (8.66E-15)	0.8480 (8.66E-15)	0.8480 (8.66E-15)	0.8480 (8.66E-15)	0.8480 (8.66E-15)	0.3594 (3.84E-02)	0.2574 (2.58E-03)	0.1886 (1.68E-02)
	$\delta_{dist}^{\lambda_B}$ (ID ² C-EDA)	0	0	0	0	0	0	+0.0197	+0.0001	+0.0006

Table 9 (Continued)

(Long, Mid)		$IP_B = 15, RP_B = 60, \tau_B = 100; IP_S = 85, RP_S = 40, \tau_S = 50$								
λ_S	λ_S^{PS-opt}	(0, 5.0000]	(0, 5.0000]	(0, 5.0000]	(0, 5.0000]	(0, 5.0000]	(0, 5.0000]	(0, 0.4374]	(0, 0.2251]	(0, 0.1296]
	$\bar{\lambda}_S^{PS-opt}$ (S-EDA)	1.7414 (1.10E+00)	1.8784 (1.12E+00)	2.4064 (1.11E+00)	1.7518 (1.06E+00)	1.8795 (1.12E+00)	2.3324 (1.09E+00)	0.9700 (3.09E-01)	1.5103 (1.33E+00)	2.1217 (1.35E+00)
	$\delta_{dist}^{\lambda_B}$ (S-EDA)	-3.2586	-3.1216	-2.5936	-3.2482	-3.1205	-2.6676	+0.5326	+1.2852	+1.9921
	$\bar{\lambda}_S^{PS-opt}$ (ID ² C-EDA)	5.0000 (0.00E+00)	5.0000 (0.00E+00)	5.0000 (0.00E+00)	5.0000 (0.00E+00)	5.0000 (0.00E+00)	5.0000 (0.00E+00)	0.7325 (2.21E-01)	0.2283 (9.60E-03)	0.1441 (4.87E-06)
	$\delta_{dist}^{\lambda_S}$ (ID ² C-EDA)	0	0	0	0	0	0	+0.2951	+0.0032	+0.0145
Convergence of EDA										
N^{Gen}	\bar{N}^{Gen} (S-EDA)	20.80	20.56	20.05	20.79	20.84	20.42	135.65	2500	2500
	\bar{N}^{Gen} (ID ² C-EDA)	68.47	69.45	68.61	70.48	70.59	70.47	934.36	160.75	181.35
N^{Re_init}	\bar{N}^{Re_init} (S-EDA)	0	0	0	0	0	0	0	10	10
	\bar{N}^{Re_init} (ID ² C-EDA)	0	0	0	0	0	0.012	0.144	0	0.002

Table 10 Results of Type-II experiments in (Mid, Short)

(Mid, Short)		$IP_B = 15, RP_B = 60, \tau_B = 50; IP_S = 85, RP_S = 40, \tau_S = 25$								
Combinations of Weights										
(w_{NP}^B, w_{NS}^B)		(0.7, 0.3)			(0.5, 0.5)			(0.3, 0.7)		
dP_c^B		28.5			37.5			46.5		
dT_c^B		35			25			15		
(w_{NP}^S, w_{NS}^S)		(0.7, 0.3)	(0.5, 0.5)	(0.3, 0.7)	(0.7, 0.3)	(0.5, 0.5)	(0.3, 0.7)	(0.7, 0.3)	(0.5, 0.5)	(0.3, 0.7)
dP_c^B		71.5	62.5	53.5	71.5	62.5	53.5	71.5	62.5	53.5
dT_c^B		14	10	6	14	10	6	14	10	6
PS-Optimization mode		(Mode 1)	(Mode 2)	(Mode 3)	(Mode 4)	(Mode 5)	(Mode 6)	(Mode 7)	(Mode 8)	(Mode 9)
Agreement Price										
P_c	P_c^{PS-opt}	40.00	40.00	40.00	40.00	40.00	40.00	46.50	46.50	46.50
	\bar{P}_c^B (S-EDA)	40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	48.60 (4.47E+00)	48.62 (5.18E+00)	47.12 (4.40E+00)
	$\delta_{dist}^{P_c^B}$ (S-EDA)	0	0	0	0	0	0	2.10	2.12	0.62
	\bar{P}_c^B (ID ² C-EDA)	40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	46.66 (4.05E-01)	46.53 (1.62E-01)	46.55 (2.83E-01)
	$\delta_{dist}^{P_c^B}$ (ID ² C-EDA)	0	0	0	0	0	0	0.16	0.03	0.05
	\bar{P}_c^S (S-EDA)	40.24 (4.33E-01)	40.31 (5.78E-01)	41.31 (8.03E-01)	40.10 (2.06E-01)	40.10 (2.00E-01)	40.70 (3.46E-01)	51.50 (4.98E+00)	51.06 (5.90E+00)	49.75 (5.47E+00)
	$\delta_{dist}^{P_c^S}$ (S-EDA)	0.24	0.31	1.31	0.10	0.10	0.70	5.00	4.56	3.25
	\bar{P}_c^S (ID ² C-EDA)	40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	46.66 (4.05E-01)	46.53 (1.62E-01)	46.55 (2.83E-01)
	$\delta_{dist}^{P_c^S}$ (ID ² C-EDA)	0	0	0	0	0	0	0.16	0.03	0.05
Agreement Time										
T_c	T_c^{PS-opt}	[1, 20.00]	[1, 20.00]	[1, 20.00]	[1, 20.00]	[1, 20.00]	[1, 20.00]	[1, 14.00]	[1, 10.00]	[1, 6.00]
	\bar{T}_c^B (S-EDA)	20.00 (0.00E+00)	20.00 (0.00E+00)	20.00 (0.00E+00)	20.00 (0.00E+00)	20.00 (0.00E+00)	20.00 (0.00E+00)	16.91 (2.12E+00)	16.14 (4.22E+00)	16.90 (4.29E+00)
	$\delta_{dist}^{T_c^B}$ (S-EDA)	0	0	0	0	0	0	+2.91	+6.14	+10.9

Table 10 (Continued)

(Mid, Short)		$IP_B = 15, RP_B = 60, \tau_B = 50; IP_S = 85, RP_S = 40, \tau_S = 25$								
\bar{T}_c^B (ID ² C-EDA)		20.00 (0.00E+00)	20.00 (0.00E+00)	20.00 (0.00E+00)	20.00 (0.00E+00)	20.00 (0.00E+00)	20.00 (0.00E+00)	15.07 (6.06E-01)	10.93 (1.08E+00)	6.82 (1.58E+00)
$\delta_{dist}^{T_c^B}$ (ID ² C-EDA)		0	0	0	0	0	0	+1.07	+0.93	+0.82
\bar{T}_c^S (S-EDA)		20.00 (0.00E+00)	20.00 (0.00E+00)	20.00 (0.00E+00)	20.00 (0.00E+00)	20.00 (0.00E+00)	20.00 (0.00E+00)	16.55 (2.02E+00)	15.27 (3.93E+00)	16.20 (4.48E+00)
$\delta_{dist}^{T_c^S}$ (S-EDA)		0	0	0	0	0	0	+2.55	+5.27	+10.2
\bar{T}_c^S (ID ² C-EDA)		20.00 (0.00E+00)	20.00 (0.00E+00)	20.00 (0.00E+00)	20.00 (0.00E+00)	20.00 (0.00E+00)	20.00 (0.00E+00)	14.33 (8.14E-01)	10.35 (1.15E+00)	6.30 (1.59E+00)
$\delta_{dist}^{T_c^S}$ (ID ² C-EDA)		0	0	0	0	0	0	+0.33	+0.35	+0.30
Strategy										
λ_B	λ_B^{PS-opt}	(0, 0.6415]	(0, 0.6415]	(0, 0.6415]	(0, 0.6415]	(0, 0.6415]	(0, 0.6415]	(0, 0.2802]	(0, 0.2216]	(0, 0.1682]
	$\bar{\lambda}_B^{PS-opt}$ (S-EDA)	0.6243 (9.37E-03)	0.6242 (9.51E-03)	0.6244 (9.46E-03)	0.6233 (9.65E-03)	0.6232 (9.63E-03)	0.6228 (9.61E-03)	0.2708 (1.29E-01)	0.2767 (1.73E-01)	0.3110 (1.52E-01)
	$\delta_{dist}^{\lambda_B}$ (S-EDA)	-0.0172	-0.0173	-0.0171	-0.0182	-0.0183	-0.0187	-0.0094	+0.0551	+0.1428
	$\bar{\lambda}_B^{PS-opt}$ (ID ² C-EDA)	0.6415 (8.44E-15)	0.6415 (8.44E-15)	0.6415 (8.44E-15)	0.6415 (8.44E-15)	0.6415 (8.44E-15)	0.6415 (8.44E-15)	0.2808 (1.26E-02)	0.2235 (1.64E-02)	0.1712 (2.31E-02)
	$\delta_{dist}^{\lambda_B}$ (ID ² C-EDA)	0	0	0	0	0	0	+0.0006	+0.0019	+0.0030
λ_S	λ_S^{PS-opt}	(0, 5.0000]	(0, 5.0000]	(0, 5.0000]	(0, 5.0000]	(0, 5.0000]	(0, 5.0000]	(0, 0.4374]	(0, 0.2251]	(0, 0.1296]
	$\bar{\lambda}_S^{PS-opt}$ (S-EDA)	2.0026 (1.17E+00)	2.2664 (1.20E+00)	2.4863 (1.09E+00)	1.9527 (1.16E+00)	2.2258 (1.24E+00)	2.4636 (1.14E+00)	2.0629 (1.61E+00)	1.6843 (1.43E+00)	2.1043 (1.57E+00)
	$\delta_{dist}^{\lambda_S}$ (S-EDA)	-2.9974	-2.7336	-2.5137	-3.0473	-2.7742	-2.5364	+1.6255	+1.4592	+1.9747
	$\bar{\lambda}_S^{PS-opt}$ (ID ² C-EDA)	5.0000 (0.00E+00)	5.0000 (0.00E+00)	5.0000 (0.00E+00)	5.0000 (0.00E+00)	5.0000 (0.00E+00)	5.0000 (0.00E+00)	0.5429 (5.50E-01)	0.2813 (4.34E-01)	0.1723 (3.45E-01)
	$\delta_{dist}^{\lambda_S}$ (ID ² C-EDA)	0	0	0	0	0	0	+0.1055	+0.0562	+0.0427
Convergence of EDA										
N^{Gen}	\bar{N}^{Gen} (S-EDA)	46.69	48.63	46.25	48.15	48.87	49.61	2500	2500	2500
	\bar{N}^{Gen} (ID ² C-EDA)	89.33	92.04	95.71	90.96	90.96	93.16	178.91	168.81	131.09
N^{Re_init}	\bar{N}^{Re_init} (S-EDA)	0	0	0	0	0	0	10	10	10
	\bar{N}^{Re_init} (ID ² C-EDA)	0	0	0	0	0	0	0.042	0.003	0.001

Table 11 Results of Type-II experiments in (Long, Short)

(Long, Short)		$IP_B = 15, RP_B = 60, \tau_B = 100; IP_S = 85, RP_S = 40, \tau_S = 25$								
Combinations of Weights										
(w_{NP}^B, w_{NS}^B)		(0.7, 0.3)			(0.5, 0.5)			(0.3, 0.7)		
dP_c^B		28.5			37.5			46.5		
dT_c^B		70			50			30		
(w_{NP}^S, w_{NS}^S)		(0.7, 0.3)	(0.5, 0.5)	(0.3, 0.7)	(0.7, 0.3)	(0.5, 0.5)	(0.3, 0.7)	(0.7, 0.3)	(0.5, 0.5)	(0.3, 0.7)
dP_c^B		71.5	62.5	53.5	71.5	62.5	53.5	71.5	62.5	53.5
dT_c^B		14	10	6	14	10	6	14	10	6
PS-Optimization mode		(Mode 1)	(Mode 2)	(Mode 3)	(Mode 4)	(Mode 5)	(Mode 6)	(Mode 7)	(Mode 8)	(Mode 9)
Agreement Price										
P_c	P_c^{PS-opt}	40.00	40.00	40.00	40.00	40.00	40.00	46.50	46.50	46.50

Table 11 (Continued)

(Long, Short)		$IP_B = 15, RP_B = 60, \tau_B = 100; IP_S = 85, RP_S = 40, \tau_S = 25$								
\bar{P}_c^B (S-EDA)		36.76 (1.52E+01)	34.00 (1.62E+01)	38.88 (1.40E+0)	38.11 (1.32E+01)	34.44 (1.64E+01)	39.52 (1.19E+01)	36.03 (2.02E+01)	42.90 (1.32E+01)	39.64 (1.73E+01)
$\delta_{dist}^{P^B}$ (S-EDA)		3.24	6.00	1.12	1.89	5.56	0.48	10.47	3.60	6.86
\bar{P}_c^B (ID ² C-EDA)		40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	46.62 (2.44E-01)	46.49 (4.91E-01)	46.54 (1.02E+00)
$\delta_{dist}^{P^B}$ (ID ² C-EDA)		0	0	0	0	0	0	0.12	0.01	0.04
\bar{P}_c^S (S-EDA)		39.76 (1.72E+01)	37.29 (1.85E+01)	42.57 (1.57E+01)	40.79 (1.48E+01)	36.72 (1.80E+01)	42.52 (1.33E+01)	38.63 (2.18E+01)	46.20 (1.48E+01)	42.46 (1.87E+01)
$\delta_{dist}^{P^S}$ (S-EDA)		0.24	2.71	2.57	0.79	3.28	2.52	7.87	0.3	4.04
\bar{P}_c^S (ID ² C-EDA)		40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	40.00 (0.00E+00)	46.62 (2.44E-01)	46.49 (4.91E-01)	46.54 (1.02E+00)
$\delta_{dist}^{P^S}$ (ID ² C-EDA)		0	0	0	0	0	0	0.12	0.01	0.04
Agreement Time										
T_c	T_c^{PS-opt}	[1, 20.00]	[1, 20.00]	[1, 20.00]	[1, 20.00]	[1, 20.00]	[1, 20.00]	[1, 14.00]	[1, 10.00]	[1, 6.00]
	\bar{T}_c^B (S-EDA)	16.89 (6.88E+00)	16.27 (7.66E+00)	17.08 (6.18E+00)	17.57 (5.95E+00)	16.20 (7.61E+00)	18.13 (5.26E+00)	14.33 (8.08E+00)	17.23 (5.38E+00)	15.64 (6.92E+00)
	$\delta_{dist}^{T^B}$ (S-EDA)	-3.11	-3.73	-2.92	-2.43	-3.80	-1.87	+0.33	+7.23	+9.64
	\bar{T}_c^B (ID ² C-EDA)	20.00 (0.00E+00)	20.00 (0.00E+00)	20.00 (0.00E+00)	20.00 (0.00E+00)	20.00 (0.00E+00)	20.00 (0.00E+00)	16.34 (2.26E+00)	12.36 (3.69E+00)	9.01 (5.56E+00)
	$\delta_{dist}^{T^B}$ (ID ² C-EDA)	0	0	0	0	0	0	+2.34	+2.36	+3.01
	\bar{T}_c^S (S-EDA)	15.34 (6.72E+00)	15.02 (7.74E+00)	15.90 (6.16E+00)	16.79 (5.89E+00)	15.26 (7.45E+00)	16.78 (5.50E+00)	12.54 (7.93E+00)	15.67 (5.43E+00)	14.57 (7.02E+00)
	$\delta_{dist}^{T^S}$ (S-EDA)	-4.66	-4.98	-4.10	-3.21	-4.74	-3.22	-1.46	+5.67	+8.57
	\bar{T}_c^S (ID ² C-EDA)	20.00 (0.00E+00)	20.00 (0.00E+00)	20.00 (0.00E+00)	20.00 (0.00E+00)	20.00 (0.00E+00)	20.00 (0.00E+00)	15.51 (3.49E-01)	11.83 (3.70E+00)	8.49 (5.30E+00)
	$\delta_{dist}^{T^S}$ (ID ² C-EDA)	0	0	0	0	0	0	+1.51	+1.83	+2.49
Strategy										
λ_B	λ_B^{PS-opt}	(0, 0.9458]	(0, 0.7481]	(0, 0.5678]	(0, 0.5445]	(0, 0.4307]	(0, 0.3731]	(0, 0.2802]	(0, 0.2519]	(0, 0.3731]
	$\bar{\lambda}_B^{PS-opt}$ (S-EDA)	0.5873 (7.88E-01)	0.7223 (9.38E-01)	0.5300 (8.32E-01)	0.5840 (8.72E-01)	0.6036 (7.27E-01)	0.5019 (7.97E-01)	0.7185 (1.07E+00)	0.3983 (7.23E-01)	0.7241 (1.27E+00)
	$\delta_{dist}^{\lambda_B}$ (S-EDA)	-0.3585	-0.0258	-0.0378	+0.0395	+0.1729	+0.1288	+0.4383	+0.1464	+0.3510
	$\bar{\lambda}_B^{PS-opt}$ (ID ² C-EDA)	0.3652 (1.89E-15)	0.3652 (1.89E-15)	0.3652 (1.89E-15)	0.3652 (1.89E-15)	0.3652 (1.89E-15)	0.3652 (1.89E-15)	0.1883 (1.34E-02)	0.1654 (2.39E-02)	0.1424 (3.85E-02)
	$\delta_{dist}^{\lambda_B}$ (ID ² C-EDA)	-0.5806	-0.3829	-0.2026	-0.1793	-0.0655	-0.0079	-0.0919	-0.0865	-0.2307
λ_S	λ_S^{PS-opt}	(0, 0.4763]	(0, 0.2451]	(0, 0.1411]	(0, 1.1809]	(0, 0.6077]	(0, 0.2962]	(0, 2.1243]	(0, 1.0000]	(0, 0.2962]
	$\bar{\lambda}_S^{PS-opt}$ (S-EDA)	1.7324 (1.25E+00)	2.2494 (1.32E+00)	1.9806 (1.15E+00)	1.9972 (1.19E+00)	1.9617 (1.37E+00)	2.0524 (1.41E+00)	2.1617 (1.42E+00)	2.1436 (1.42E+00)	2.3895 (1.49E+00)
	$\delta_{dist}^{\lambda_S}$ (S-EDA)	+1.2561	+2.0043	+1.8395	+0.8163	+1.3540	+1.7562	+0.0374	+1.1436	+2.0933
	$\bar{\lambda}_S^{PS-opt}$ (ID ² C-EDA)	5.0000 (0.00E+00)	5.0000 (0.00E+00)	5.0000 (0.00E+00)	5.0000 (0.00E+00)	5.0000 (0.00E+00)	5.0000 (0.00E+00)	1.3152 (1.63E+00)	0.9042 (1.49E+00)	0.6914 (1.15E+00)
	$\delta_{dist}^{\lambda_S}$ (ID ² C-EDA)	+4.5237	+4.7549	+4.8589	+3.8191	+4.3923	+4.7038	-0.8091	-0.0958	+0.3952
Convergence of EDA										
N^{Gen}	\bar{N}^{Gen} (S-EDA)	1976.16	1893.64	1829.05	1881.40	1943.57	1844.05	2500	2500	2500
	\bar{N}^{Gen} (ID ² C-EDA)	176.98	183.82	190.06	186.18	196.66	196.67	235.57	237.53	181.80
N^{Re_init}	\bar{N}^{Re_init} (S-EDA)	7.568	6.964	7.254	7.381	6.527	7.091	10	10	10
	\bar{N}^{Re_init} (ID ² C-EDA)	0	0	0	0	0	0	0.001	0	0.039

Analysis From the results of optimization modes 1 to 6 in Tables 9 to 11, it can be observed that the values of $\bar{N}^{Re_init(S-EDA)}$ were 0 in (Long, Mid) and (Mid, Short) but were very high (> 6) in (Long, Short). Hence, S-EDAs required an extremely many generations for coevolving converged populations in (Long, Short). This is because (Long, Short) has a large search space compared to (Long, Mid) and (Mid, Short). These indicate that the search capability of S-EDAs was deteriorated in the large search space of (Long short). In contrast, it can also be observed from the results of optimization modes 1 to 6 in Tables 9 to 11 that the values of $\bar{N}^{Re_init(ID^2C-EDA)}$ were all 0. This indicates that ID²C-EDAs have enough search capability even in the large search space. From most of the results of the optimization modes 7 to 9 in Tables 9 to 11, it can be observed that the values of $\bar{N}^{Re_init(S-EDA)}$ were 10. This indicates that S-EDAs could not have enough capability for coevolving converged populations within reasonable numbers of generations for these modes. In contrast, from all the results of the optimization modes 7 to 9 in Tables 9 to 11, it can be observed that the values of $\bar{N}^{Re_init(ID^2C-EDA)}$ were very small. This indicates that ID²C-EDAs have enough capability for coevolving converged populations within reasonable numbers of generations for these modes. A similar analysis used in Observation 2 can be used to explain why ID²C-EDAs outperformed S-EDAs.

From the Observations 3 to 5, we can draw the following conclusion for the Type-II experiments.

Conclusion 2 When the negotiation settings with insufficient AgZ^{NP} are provided for PS-optimizing agents, ID²C-EDAs generally coevolve effective converged PS-optimizing negotiation strategies for both B and S within reasonable numbers of generations while S-EDA has a high possibility of failure in coevolving such negotiation strategies.

6 Related works

Since this work mainly focuses on finding effective negotiation strategies for PS-optimizing agents with incomplete information using coevolutionary learning, the related works are approaches using EAs for evolving negotiation strategies.

There are some existing works on using EAs as a decision making component to determine an agent's optimal negotiation strategy (that ensures reaching an agreement successfully and achieving higher utilities) under incomplete information settings by generating adaptive proposals at every negotiation round (e.g., [18, 39, 40]). However, EAs in this work were used for learning agents' negotiation strategies to find both agents' effective negotiation strategies through coevolutionary learning under an incomplete information set-

ting. Hence, this section only introduces and discusses related works on applying EAs to learn effective negotiation strategies.

In [24], Oliver has utilized standard GAs [7] for learning (simple) strategies of agents in which agents use quite simple threshold rules for bargaining and showed that by adopting GAs, agents can learn strategies for simple negotiation games. Whereas empirical studies seem to indicate the agents in [24] are generally successful in learning effective strategies, the research is only limited to learning simple threshold rules; offers are accepted if agents learn strategies having a higher utility over a predefined threshold.

In [23], Matos et al. have also utilized a GA for learning the most successful strategies against different types of opponents in different negotiation situations in which a service-oriented negotiation model in [4] was adopted to determine successful strategies for different types of environment by coevolving negotiation strategies and tactics. Empirical results in [23] are carried out for bilateral negotiations having two issues and showed that the agents adopting the GA are generally effective in evolving effective strategies for different negotiation circumstances. Nevertheless, the approach used in [23] has a serious limitation in that it requires a centralized coevolution model where complete information about each agent is assumed for evolving populations using one GA. Furthermore, such assumption is not realistic in many practical negotiation systems in which agents generally have incomplete information about each other.

In [13], Jin and Tsang have utilized a genetic programming (GP) for comparing the evolved results achieved from the GP with sub-game perfect equilibrium (SPE) solutions from game-theoretic analysis for complete information bargaining problems and showed that GP results achieved approximate solutions to the SPE solutions. Later, in [12], Jin has extended the simple bargaining problems to incomplete information bargaining problems and showed that the GP was capable of achieving reasonably good solutions. Similar to [23], the works [13] and [12] also have the problem of the centralized coevolution model for the coevolutionary learning using one GP.

This paper significantly and considerably extended the previous works reported in [8, 9, 32] and [10].

In [32], Sim has utilized an EDA (specifically, UMDA_c) for coevolving effective negotiation strategies of agents having difference preference criteria for optimizing price and optimizing negotiation speed and it seems that the preliminary empirical results in [32] showed that the EDA was capable of coevolving PS-optimizing negotiation strategies of agents for P-optimizing negotiation. The fitness function used in [32] was the (total) utility function, which is similar to (7) in this work, consisting of both price and speed utility functions in which each weighting factor was incorporated

to its corresponding utility function. However, using the co-evolved results in [32], it is difficult to investigate convergence of the EDA because the results have no information about generations required for achieving converged populations. Later, in the extended work [9], Gwak and Sim have found the problem of the fitness function in [32] and have devised new fitness functions based on measuring the difference between: (1) the ratio of the price weighting factor to the time weighting factor and (2) the corresponding ratio of price utility to speed utility. Although the fitness function in [32] has some ambiguity in defining better negotiation solutions with higher fitness in the composite utility space (as described in Sect. 4.1), the fitness functions in [9] are more effective in defining better negotiation solutions. This is because they can differentiate negotiation strategies with the given ratio of the price weighting to the speed weighting from others. Empirical results in [9] showed that the devised fitness functions outperform the fitness function used in [32]. Furthermore, comparing the coevolution performance between conventional GA and EDA for finding effective negotiation strategies, it can be found that both conventional GA and EDA have limited performance for co-evolving effective negotiation strategies.

In [8] and [10], Gwak and Sim empirically have proved that conventional GA and EDA generally could not achieve effective (or near-optimal) coevolution results for finding optimal P-optimizing negotiation solutions. In addition, under the assumption that dynamic diversity controlling methods can assist EAs to coevolve optimal solutions for both populations (in which both agents adopted different EAs and a decentralized coevolution model was assumed), the DR procedure was devised and two local improvement methods called LNS and PR were also devised for further coevolution performance improvement. If DR together with LNS and PR is incorporated with conventional GA (respectively, conventional EDA), we called it ID²C-GA (respectively, ID²C-EDA). Empirical results showed that ID²C-GA and ID²C-EDA have complementary performance in that one achieved better performance for some cases and the other one achieved better performance for the other cases. Furthermore, since it was also shown that ID²C-EDA has better performance than ID²C-GA for coevolving optimal strategies in the larger solution space, we adopted ID²C-EDA as the EA model for the coevolutionary learning.

Finally, it is acknowledged that this work significantly and considerably enhances [32] and [9] as well as the closely related works [12, 13, 23, 24] as follows:

1) In [32] and [9], there is theory only for optimal negotiation solutions of P-optimizing negotiations given as Theorems 1 and 2; however, there is no such theory for the other types of negotiations (e.g., for PS-optimizing negotiations). To this end, this work provides theoretical background of optimal negotiation solutions

for PS-optimizing negotiations given as Theorems 3 to 5 by designing optimal PS-optimizing agents with complete information. Hence, it is possible (i) to calculate optimal negotiation solutions for each PS-optimization mode and (ii) to evaluate optimality of coevolved negotiation solutions (under an incomplete information setting) by comparing them with the optimal negotiation solutions.

- (2) Since the fitness function in [9] (showing better performance than the fitness function in [32]) simply measures difference of ratios between weightings and utilities, higher fitness values indicate that the ratios are (much) closer. Furthermore, using the fitness function in [25], it is hard to find direct relationship between fitness and optimal PS-optimizing negotiation solutions achieved from Theorems 3 to 5. Hence, it is required to develop the new fitness function in (17) which is based on composite likelihoods of agreement price in (14) and agreement time in (15).
- (3) Although the previous works [12, 13, 23, 24] and [32] assumed a fully centralized coevolution model using one EA in which complete information about each agent is assumed for evolutionary learning, this work provides the fully decentralized coevolution model in Sect. 4.2 using two EDAs (for populations of B and S , respectively) and a coordinator to share and determine conditions of coevolution termination and inappropriate coevolution. The decentralized coevolution model will be more realistic in simulating negotiations of agents having fully incomplete information about each other.
- (4) Although the experiments in both [32] and [9] were carried out under the assumption that both agents have the same negotiation mode (e.g., if B is *exact*-PS-optimizing, then S is also *exact*-PS-optimizing), this work carried out extensive experiments by considering all possible combinations of (representative) PS-optimizing negotiation modes between B and S (see Table 3).
- (5) As shown in [8] and [10], conventional EAs such as S-GAs and S-EDAs has some drawbacks for coevolutionary learning due to premature convergence and biased coevolution effects, which can be also observed from the results in [32] and [9]. In contrast, ID²C-GAs and ID²C-EDAs have enough coevolution capability for coevolutionary learning because they are augmented with DR together with LNS and PR. Empirical results in Observations 1 to 5 demonstrated that effective (sometimes optimal) PS-optimizing negotiation solutions can be achieved using ID²C-EDAs for the coevolutionary learning while such solutions cannot be achieved using S-EDAs. Hence, this paper can be seen as an extension of [8] and [10] providing empirical evidences of the effectiveness of ID²C-EDAs for coevolutionary learning.

7 Conclusion and future work

Based on the theoretical results obtained in Sect. 4.1 for finding optimal negotiation strategies of PS-optimizing agents with complete information, this work has developed an effective coevolutionary learning mechanism (by adopting ID²C-EDAs) for finding effective PS-optimizing negotiation strategies of PS-optimizing agents with incomplete information. The novel feature and significance of this research is therefore designing and developing negotiation mechanisms that can: (1) optimize both price and negotiation speed of PS-optimizing agents with complete information and (2) co-evolve effective, or (near-)optimal, negotiation strategies for PS-optimizing agents with incomplete information.

The contributions of this work are detailed as follows.

- (1) For PS-optimizing negotiation under a complete information setting, this work determines P_c^{PS-opt} and T_c^{PS-opt} (Theorems 3 to 5 in Sect. 4.1) that lead to optimal negotiation strategies for both PS-optimizing agents [(12) and (13) in Sect. 4.1]. Whereas Theorems 1 and 2 are based on Theorems 1 and 2 in [40, pp. 199–200], this research, to the best of the authors’ knowledge, is the earliest work suggesting optimality of agreements between PS-optimizing agents with incomplete information.
- (2) This contribution distinguishes this work from [40] and [5] in that (i) [5] only showed that there are three classes of optimal strategy such as *Boulware*, *Linear* and *Conceder* depending on different negotiation scenarios and (ii) [40] only focused on showing that there is optimal negotiation strategies for both P-optimizing agents in which one agent (having a bargaining advantage over the opponent in terms of time) maximizes its price utility and guarantees that an agreement is reached. The following summarizes the optimality of agreements for P-optimizing and PS-optimizing agents.

Optimal agreement point of P-optimizing agents

$$\text{When } \tau_B > \tau_S \begin{cases} P_c^{P-opt} = RP_S \\ T_c^{P-opt} = \tau_S \end{cases}$$

$$\text{When } \tau_B < \tau_S \begin{cases} P_c^{P-opt} = RP_B \\ T_c^{P-opt} = \tau_B \end{cases}$$

Optimal agreement points of PS-optimizing agents

$$\text{When } \tau_B > \tau_S \begin{cases} P_c^{PS-opt} = \max_{dP_c^x} \{U_{NP}^B(dP_c^B), U_{NP}^B(dP_c^S)\} \\ T_c^{PS-opt} = [0, \max_{dT_c^x} \{U_{NS}^B(dT_c^B), U_{NS}^B(dT_c^S)\}] \end{cases}$$

$$\text{When } \tau_B < \tau_S \begin{cases} P_c^{PS-opt} = \max_{dP_c^x} \{U_{NP}^B(dP_c^B), U_{NP}^B(dP_c^S)\} \\ T_c^{PS-opt} = [0, \max_{dT_c^x} \{U_{NS}^B(dT_c^B), U_{NS}^B(dT_c^S)\}] \end{cases}$$

- (3) Whereas several existing works (discussed in Sect. 6) adopt EAs for evolving successful negotiation strategies for agents under different negotiation situations, these works are limited in that: (1) agents mostly did not consider optimization of both price and negotiation speed and (2) centralized coevolution models were used for coevolutionary learning in which complete information settings for agents are generally assumed. However, agents in this work are designed to optimize both price and negotiation speed using coevolutionary learning for an incomplete information setting. Furthermore, we adopted a decentralized coevolution model in which incomplete information settings can be generally assumed.
- (4) In comparison with authors’ previous works [32] and [9], this paper has provided much more detailed and enhanced designs for PS-optimizing agents for both complete and incomplete information settings (Sect. 4).
- (5) A new fitness function was designed and implemented for S-EDAs and ID²C-EDAs and has the following novel features.
 - (a) The likelihood based (Gaussian) distance metric was formulated and applied to measure the closeness between the achieved agreement price and time and the desired agreement price and time, respectively.
 - (b) The fitness function is composed of a weighted linear combination of individual similarities for price and negotiation speed in which each similarity is weighted by the corresponding preference weight and is also magnified by the weight for further discrimination from the other similarities.
 - (c) While the previous fitness functions in [32] and [9] as the form of utility functions lead to the deterioration of intensification capability of both EDAs (Sect. 4.2), the proposed fitness function is effective in coevolving effective negotiation strategies for both PS-optimizing agents.
- (6) Empirical results (Observations 1 to 5) show that (i) ID²C-EDAs significantly outperforms S-EDAs in terms of coevolution performance for achieving close to the optimum and balanced solutions and (ii) throughout the (decentralized) coevolutionary learning, ID²C-EDAs adopting the proposed fitness function generally achieved effective, or (near-)optimal, negotiation strategies for both PS-optimizing agents for various combinations of preference criteria under the negotiation settings having both sufficient and insufficient AgZ^{NP} . From these results, we can conclude that ID²C-EDAs are more suitable than S-EDAs for the coevolutionary learning in achieving effective PS-optimizing negotiation strategies. Hence, this work together with [10]

can also provide evidences to show the effectiveness of ID²C-EDAs for competitive coevolution of heterogeneous populations.

Finally, the authors acknowledge that although this work develops a coevolutionary learning approach for finding effective PS-optimizing negotiation strategies under an incomplete information setting in which one agent having a bargaining advantage over the other in terms of time, this work in its present form does not deal with the negotiation situation that neither agent has bargaining advantage in terms of time. Hence, extending this work to include the design of PS-optimizing agents neither having a bargaining advantage is on the agenda for future work. Since the focus of this work is designing a PS-optimizing negotiation mechanism, for simplicity, this work only considers bilateral single-issue negotiations between PS-optimizing agents. In addition, ID²C-EDAs were adopted for coevolutionary learning because they have showed better performance in searching larger solution space than ID²C-GAs for competitive coevolution [10]. Therefore, other possible enhancements of this work may include: (1) extending this work to deal with multi-issue negotiations and (2) adopting other possible types of EAs with the dynamic diversity controlling method in this work to compare the coevolution performances and find a more effective and efficient EA model for the coevolutionary learning.

References

- Ahn CW, Ramakrishna RS (2008) On the scalability of real-coded Bayesian optimization algorithm. *IEEE Trans Evol Comput* 12(3):307–322
- Ardaiz O, Artigas P, Eymann T, Freitag F, Navarro L, Reinicke M (2006) The catalaxy approach for decentralized economic-based allocation in grid resource and service markets. *Appl Intell* 25(2):131–145
- Axelrod R (1987) The evolution of strategies in the iterated prisoner's dilemma. In: Davis L (ed) *Genetic algorithms and simulated annealing*. Morgan Kaufmann, Los Altos
- Faratin P, Sierra C, Jennings NR (1998) Negotiation decision functions for autonomous agents. *Robot Auton Syst* 24(3):159–182
- Fatima S, Wooldridge M, Jennings NR (2005) Bargaining with incomplete information. *Ann Math Artif Intell* 44(3):207–232
- Gatti N, Giunta FD, Marino S (2008) Alternating-offers bargaining with one-sided uncertain deadlines: an efficient algorithm. *Artif Intell* 172(8–9):1119–1157
- Goldberg DE (1989) *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley Longman, Reading
- Gwak J, Sim KM (2010) Novel dynamic diversity controlling EDA and its application to automated bilateral negotiation. In: *Proceedings of the IEEE 5th international conference of bio-inspired computing: theories and applications (BIC-TA'10)*, pp 536–544
- Gwak J, Sim KM (2011) Coevolving negotiation strategies for p-s-optimizing agents. In: Ao S-I, Castillo O, Huang X (eds) *Intelligent control and computer engineering*. LNEE, vol 70. Springer, Berlin, pp 119–135
- Gwak J, Sim KM Novel dynamic diversity controlling EAs for co-evolving optimal negotiation strategies. *Inf Sci* (to be submitted)
- Jennings NR, Faratin P, Lomuscio AR, Parsons S, Sierra C, Wooldridge M (2001) Automated negotiation: prospects, methods and challenges. *Group Decis Negot* 10(2):199–215
- Jin N (2005) Equilibrium selection by co-evolution for bargaining problems under incomplete information about time preferences. In: Come D, et al (eds) *Proceedings of the 2005 IEEE congress on evolutionary computation (CEC'05)*, vol 3, pp 2661–2668
- Jin N, Tsang E (2005) Co-evolutionary strategies for an alternating-offer bargaining problem. In: *Proceedings of the IEEE 2005 symposium on computational intelligence and games (CIG'05)*, pp 211–217
- Jonker CM, Robu V, Treur J (2007) An agent architecture for multi-attribute negotiation using incomplete preference information. *Auton Agents Multi-Agent Syst* 15(2):221–252
- Kraus S (2001) *Strategic negotiation in multi-agent environments*. MIT Press, Cambridge
- Lai G, Sycara K (2009) A generic framework for automated multi-attribute negotiation. *Group Decis Negot* 18:169–187
- Larrañaga P, Lozano JA (2002) *Estimation of distribution algorithms: a new tool for evolutionary computation*. Kluwer, Norwell
- Lau R (2006) An evolutionary learning approach for adaptive negotiation agents. *Int J Intell Syst* 21(1):41–72
- Lewis PR, Marrow P, Yao X (2010) Resource allocation in decentralised computational systems: an evolutionary market-based approach. *Auton Agents Multi-Agent Syst* 21(2):143–171
- Li C, Li L (2006) Multi economic agent interaction for optimizing the aggregate utility of grid users in computational grid. *Appl Intell* 25(2):147–158
- Li C, Giampapa JA, Sycara K (2006) Bilateral negotiation decisions with uncertain dynamic outside options. *IEEE Trans Syst Man Cybern, Part C, Appl Rev* 36(1):31–44
- Lomuscio A, Wooldridge M, Jennings NR (2003) A classification scheme for negotiation in electronic commerce. *Group Decis Negot* 12(1):31–56
- Matos N, Sierra C, Jennings NR (1998) Determining successful negotiation strategies: an evolutionary approach. In: *Proceedings of 3rd international conference of multi-agent systems*, pp 182–189
- Oliver JR (1996) A machine-learning approach to automated negotiation and prospects for electronic commerce. *J Manag Inf Syst* 13(3):83–112
- Pelikan M, Goldberg DE, Lobo FG (2002) A survey of optimization by building and using probabilistic models. *Comput Optim Appl* 21(1):5–20
- Rubinstein A (1985) A bargaining model with incomplete information about time preferences. *Econometrica* 53(5):1151–1172
- Schnizler B, Neumann D, Veit D, Weinhardt C (2008) Trading grid services—a multi-attribute combinatorial approach. *Eur J Oper Res* 187(3):943–961
- Sim KM (2002) A market-driven model for designing negotiation agents. *Comput Intell* 18(4):618–637
- Sim KM (2005) Equilibria, prudent compromises, and the “waiting” game. *IEEE Trans Syst Man Cybern, Part B, Cybern* 35(4):712–724
- Sim KM (2006) G-commerce, market-driven G-negotiation agents and grid resource management. *IEEE Trans Syst Man Cybern, Part B, Cybern* 36(6):1381–1394
- Sim KM (2006) Guest editorial: agent-based grid computing. *Appl Intell* 25(2):127–129
- Sim KM (2008) An evolutionary approach for p-s-optimizing negotiation. In: *Proceedings of the 2008 IEEE congress on evolutionary computation (CEC'08)*, pp 1364–1371
- Sim KM (2009) Agent-based cloud commerce. In: *Proceedings of the IEEE international conference on industrial engineering and engineering management (IEEM'09)*, pp 717–721

34. Sim KM (2010) Grid resource negotiation: survey and new directions. *IEEE Trans Syst Man Cybern, Part C, Appl Rev* 40(3):245–257
35. Sim KM (2010) Towards complex negotiation for Cloud economy. In: *Proceedings of the 5th international conference on grid and pervasive computing (GPC'10)*. LNCS, vol 6104, pp 395–406
36. Sim KM (2011) Agent-based cloud computing. *IEEE Trans Serv Comput*. doi:10.1109/TSC.2011.52 (Special Issue on Cloud Computing). http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6042853
37. Sim KM, Choi CY (2003) Agents that react to changing market situations. *IEEE Trans Syst Man Cybern, Part B, Cybern* 33(2):188–201
38. Sim KM, Shi B (2010) Concurrent negotiation and coordination for controlling grid resource co-allocation. *IEEE Trans Syst Man Cybern, Part B, Cybern* 40(2):753–766
39. Sim KM, Guo Y, Shi B (2007) Adaptive bargaining agents that negotiate optimally and rapidly. In: *Proceedings of the IEEE congress on evolutionary computation (CEC'07)*, pp 1007–1014
40. Sim KM, Guo Y, Shi B (2009) BLGAN: Bayesian learning and genetic algorithm for supporting negotiation with incomplete information. *IEEE Trans Syst Man Cybern, Part B, Cybern* 39(1):198–211
41. Venugopal S, Chu X, Buyya R (2008) A negotiation mechanism for advance resource reservation using the alternate offers protocol. In: *Proceedings 16th international workshop on quality of service (IWQoS)*, pp 40–49
42. Xing H, Qu R (2012) A compact genetic algorithm for the network coding based resource minimization problem. *Appl Intell* 36(4):809–823



Jeonghwan Gwak is a Ph.D. candidate in the School of Information and Mechatronics at the Gwangju Institute of Science and Technology, South Korea. From 2002 to 2007, he had worked for several companies and research institutes as a researcher, as well as a chief technician. His current research interests focus on evolutionary computation, coevolutionary algorithms, resource coallocation in Cloud computing and the applications of evolutionary techniques to multiagent systems.



Kwang Mong Sim is a Chair (Professor) in Computer Science at the University of Kent, Chatham Maritime, Kent, UK. Professor Sim was the Founder and Director of the Multiagent and Cloud Computing Laboratory at the Gwangju Institute of Science and Technology in South Korea. He is an Associate Editor for the *IEEE Transactions on Systems, Man and Cybernetics—Part C*, the *International Journal of Cloud Computing* and many other international journals. He is also the Guest Editor of five (IEEE) journal special issues in agent-based Grid computing and automated negotiation, including a special issue on Grid Resource Management in the *IEEE Systems Journal* (IEEE Systems Council). He served as a referee for several national research grant councils including the National Science Foundation in USA. He has delivered many keynote lectures on agent-based Cloud computing and automated negotiation in many international conferences, and has published many survey papers, technical papers, and award-winning conference papers in Cloud computing and multiagent systems. He received his PhD and MSc from the University of Calgary, AB, Canada, and graduated Summa Cum Laude with a B.Sc. (Hon) from the University of Ottawa, ON, Canada.