

# Coevolving Negotiation Strategies for P-S-Optimizing Agents

Jeonghwan Gwak and Kwang Mong Sim

**Abstract** In this paper, we consider the negotiation between two competitive agents that consider both time and cost criteria. Therefore, the negotiation agents are designed to not only optimize price utility but also be successful in optimizing (negotiation) speed utility. To this end, the objective of this work is to find effective strategies for the negotiation. The strategies are coevolved through an evolutionary learning process using two different evolutionary algorithms (EAs)—a genetic algorithm (GA) and an estimation of distribution algorithm (EDA). We present an empirical comparison of GA and EDA in coevolving negotiation strategies with different preference criteria in optimizing the price and (negotiation) speed. The experimental results show that both EAs are successful in finding good solutions with respect to both the price-optimizing (*P-Optimizing*) and the speed-optimizing (*S-Optimizing*) negotiation. However, both EAs are not effective in the negotiation for the concurrent optimization of the price and speed (*P-S-Optimizing* negotiation). This is because in some cases, the original fitness function cannot characterize the difference among *P-Optimizing*, *S-Optimizing*, and *P-S-Optimizing* solutions. Hence, this paper proposes a new fitness function that can better differentiate among the *P-Optimizing*, *S-Optimizing*, and *P-S-Optimizing* solutions. The experiments showed that the EAs using the proposed fitness function can coevolve effective strategies for the exact *P-S-Optimizing* negotiation.

**Keywords** Software agent · Price and negotiation speed concurrent optimizing negotiation · Genetic algorithms · Estimation of distribution algorithms

---

K.M. Sim (✉)

School of Information and Communications, Gwangju Institute of Science and Technology (GIST), Gwangju 500-712, South Korea  
e-mail: [kmsim@gist.ac.kr](mailto:kmsim@gist.ac.kr)

## 1 Introduction

Automated negotiation is a process in which a group of agents communicate with each other either directly or indirectly to resolve their differences in the hope of eventually reaching an agreement. The interacting agents negotiate to coordinate their activities and to reach mutually acceptable agreements about the division of labor and resources [1].

For a Grid [2] to efficiently support a variety of applications, a resource management system is central to its operation [3]. Grid resource management [3, 4] involves multiple criteria optimization, and some of these criteria are generally classified into time criteria and cost criteria [5]. Sim [6–9] argued and showed that negotiation agents can play an essential role in realizing the Grid vision because the agents can act flexibly for the Grid resources whose performance changes dynamically. To maintain the good performance of the system, the negotiation agents for Grid resource management should be designed to not only optimize price utility (i.e., cost criteria) but be successful in reaching early agreements (i.e., time criteria) [4, 5]. This is because any delay in making a successful negotiation to acquire all the required Grid resources for Grid applications (i.e., resource consumers) before the deadline for executing a job will be considered as an overhead.

Different resource owners and consumers may have different objectives, policies, and preferences [5, 7]. For example, consumers may have conflicting criteria between acquiring cheaper resources and achieving faster negotiation speed (i.e., response time). The resource owner (respectively, a resource consumer) that prefers cheaper resource alternatives at the expense of having to wait longer is said to be more price optimizing (*P-Optimizing*), while a speed-Optimizing (*S-Optimizing*) resource owner (respectively, resource consumer) prefers to obtain a resource more rapidly perhaps by paying a higher price at an earlier round of negotiation than its deadline. Different emphasis on optimizing price and optimizing negotiation speed can be modeled by placing different weights on the two criteria. An exact (or equally distributed) concurrent price and speed optimizing (*P-S-Optimizing*) agent has equal emphasis on the two criteria. In this work, we use three negotiation modes for both competitive agents: *P-Optimizing*, *S-Optimizing*, and *P-S-Optimizing* negotiation.

The idea of adopting an EDA for coevolving the negotiation strategies of the agents that have different preference criteria such as optimizing price and optimizing negotiation speed was first proposed in [5]. The problem of ineffectiveness in coevolving negotiation strategies for the *P-S-Optimizing* negotiation is presented in [17]. Furthermore, in [17], one possible solution for resolving the difficulties of a *P-S-Optimizing* negotiation is suggested by restricting the solution space (which is the EA's perspective; from the agents' perspective, the negotiation corresponds to the adoption of a feasible strategy space) using predefined strategy profiles. However, in some cases, the fitness function originally used in [5] and [17] cannot effectively discriminate between the different emphases on the two criteria in the total utility space. To solve this problem, we propose a new fitness function that can better characterize the differences among *P-Optimizing*, *S-Optimizing*, and *P-S-Optimizing* solutions.

The rest of this paper is organized as follows: Section 2 specifies the negotiation model of this work. The coevolutionary framework based on GA and EDA is described in Sect. 3. The problem of coevolving strategies for P-S-Optimizing negotiation and its solution will be presented in Sect. 4. Section 5 presents the experimental results and analyses. Finally, Sect. 6 concludes this work with a summary and suggestions for future work.

## 2 P-S-Optimizing Negotiation

In a classical bargaining model, the utility function  $U^x$  of agent  $x$ , where  $x \in \{B, S\}$  and  $\hat{x}$  denotes  $x$ 's opponent, is defined as follows: Let  $IP_x$  and  $RP_x$  be the initial and the reserve prices of  $x$ . Let  $D$  be the event in which  $x$  fails to reach an agreement with its opponent  $\hat{x}$ .  $U^x : [IP_x, RP_x] \cup D \rightarrow [0, 1]$  such that  $U^x(D) = 0$  and  $U^x(P_x) > U^x(D)$  for any possible proposal  $P_x \in [IP_x, RP_x]$ . If  $x$  and  $\hat{x}$  are sensitive to time, let  $\tau_x$  be the deadline of  $x$  and  $\tau_{\hat{x}}$  be the deadline of  $\hat{x}$ . An agreement price that is acceptable to both  $B$  and  $S$  lies within the interval  $[RP_S, RP_B]$ .

In the bargaining model with complete information between  $B$  and  $S$ , both agents know the opponent's initial price, reserve price, and deadline. If one of the agents has a significantly longer deadline than its opponent, the agent that has a longer deadline will have sufficient bargaining advantages. In other words, an agent that has the longer deadline will dominate the negotiation. Under these conditions, Sim [10, 11] proved that an agent's optimal strategy can be computed using its opponent's deadline and reserve price. It can be stated as the following theorem:

**Theorem 1** *If agent  $x$ 's deadline  $\tau_x$  is significantly longer than its opponent's deadline  $\tau_{\hat{x}}$  (i.e.,  $\tau_x \gg \tau_{\hat{x}}$ ), agent  $x$  achieves its maximal utility when it adopts the strategy  $\lambda_x = \log_{\frac{\tau_x}{\tau_{\hat{x}}}} \left( \frac{RP_{\hat{x}} - IP_x}{RP_x - IP_x} \right)$ .*

### 2.1 Utility Functions

The total (aggregated) utility function  $U^x$  of agent  $x$  is composed of two attributes—price and time (i.e., the number of negotiation rounds)—and is defined as follows.

$$U^x(P_x, T_x) = w_P \times U_P^x(P_x) + w_S \times U_S^x(T_x) \quad (1)$$

where  $P_x \in \{0, P_C\}$  and  $T_x \in \{0, T_C\}$ , and  $P_C$  and  $T_C$  is are the price and time at which an agreement is reached.  $U_P^x(P_C) \in [0, 1]$  is the price utility of  $x$ , and  $U_S^x(T_C) \in [0, 1]$  is the speed utility of  $x$ .  $w_P$  and  $w_S$  are the weighting factors for price and (negotiation) speed, respectively, and  $w_P + w_S = 1$ .

The price and speed utilities are given as follows:

$$U_P^x(P_x) = \begin{cases} u_{\min}^P + (1 - u_{\min}^P) \left( \frac{RP_x - P_C}{RP_x - IP_x} \right), & \text{if an agreement is reached} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

$$U_S^x(T_x) = \begin{cases} u_{\min}^S + (1 - u_{\min}^S) \left( 1 - \frac{T_C}{\tau_x} \right), & \text{if an agreement is reached} \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where  $u_{\min}^P$  is the minimum utility when  $x$  receives a deal at its reserve price and  $u_{\min}^S$  is the minimum utility when  $x$  receives a deal at its deadline. For the experimental purpose, the values of  $u_{\min}^P$  and  $u_{\min}^S$  are defined as 0.1.

If  $x$  does not reach an agreement before its deadline,  $U_P^x = U_S^x = 0$ , and thus,  $U^x = 0$ . Otherwise,  $U^x(P_C, T_C) > 0$ .

## 2.2 Negotiation Strategies

This work considers the bilateral negotiation between  $B$  and  $S$  with incomplete information in which both agents do not know each other's deadline and reserve price. Both  $B$  and  $S$  are sensitive to time; further, we adopt the time-dependent strategies proposed in [12]. The proposal  $P_t^x$  of  $x$  at time  $t$ ,  $0 \leq t \leq \tau_x$  is defined as follows:

$$P_t^x = IP_x + (-1)^\alpha \left( \frac{t}{\tau_x} \right)^{\lambda_x} |RP_x - IP_x| \quad (4)$$

where  $\alpha = 0$  for  $B$  and  $\alpha = 1$  for  $S$ , and  $0 \leq \lambda_x \leq \infty$ .

The time-dependent strategies can be classified into three categories: (i) *conservative* (conceding slowly,  $\lambda_x > 1$ ), (ii) *linear* (conceding linearly,  $\lambda_x = 1$ ), and (iii) *conciliatory* (conceding rapidly,  $0 < \lambda_x < 1$ ) [12, 13].

## 2.3 Negotiation Protocol

The negotiation between  $B$  and  $S$  is carried out using Rubinstein's alternating offers protocol [14].  $B$  and  $S$  can conduct the negotiation only at discrete time points (e.g., in this work,  $S$  makes an offer at  $t = 0, 2, 4, 6, \dots$ , and  $B$  makes a counter-offer at  $t = 1, 3, 5, 7, \dots$ ). The negotiation process ends in both cases: (i) once an offer or a counter-offer is immediately accepted (i.e., an agreement is reached) by the other one or (ii) once the earlier deadline is reached without any agreement. In the latter case, the negotiation process ends in a conflict, and the utility outcome will be zero.

## 2.4 Objective

For the given different deadlines and different preferences of the cost and time criteria (i.e., different values of  $w_P$  and  $w_S$ ), agents will face different opponents with

different deadlines and strategies. Under these conditions, the objective of this work is to find an effective strategy  $\lambda_x$  that would optimize  $U^x(P_x, T_x)$ . In this work, (evolutionary) learning is based on two asymmetric populations in which each population has its own fitness evaluation. Agents learn effective strategies by interacting with individuals from the other population through random pairing. In the following section, the detailed procedure will be described.

### 3 Coevolutionary Models for P-S-Optimizing Negotiation

When populations between two or more species interact, each may evolve in response to the characteristics of the other. The natural coevolution refers to the mutual (or inter-dependent) evolution between interacting populations. The survival skills of the natural coevolution by making mutually beneficial arrangements have long inspired scientists to develop coevolutionary algorithms for highly dependent problems in which there are strong interactions between two elements or among several elements.

In the bilateral negotiation problem domain, inter-population based coevolution having two populations is considered. The fitness of each individual of one population depends on each individual of the other population, and hence, an individual's fitness landscape is not fixed but coupled. Therefore, coevolution is regarded as a type of landscape coupling where adaptive moves by one individual can potentially change the landscape of the other. The interaction between two populations comes from the pairing of strategies of each population, and therefore, a successful pairing mechanism is important. Furthermore, to achieve a better performance, the resulting pairing should make a sufficiently prevailing set in the feasible set. In this work, we use one-to-one random pairing because of its simplicity and efficiency.

To coevolve effective strategies under different deadlines and different weights of price and speed preferences, a coevolutionary framework using real-coded GA and EDA is implemented.  $B$  and  $S$  have each of their populations  $d^B$  and  $d^S$  consist of a set of candidate strategies.  $D^B$  and  $D^S$  are the mating pool ( $MP$ ) of  $B$  and  $S$ , respectively.

The evolution of the strategies of one population affects the strategies of the other population. In the process of coevolving strategies, each individual of the two populations will negotiate with the other through one-to-one random pairing. The fitness value of each individual is determined by its negotiation outcome. The details of the GA and EDA are as follows:

#### 3.1 Encoding Scheme

A binary coding mechanism has drawbacks because of the existence of Hamming cliffs and the lack of computation precision [15, 16]. Therefore, in the GA and EDA, each negotiation strategy of the populations is encoded as a real number. Each

**Table 1** Characterizations of GA and EDA

EA type	Method	Details
GA [17]	Selection	$k$ -tournament selection
	Crossover	Heuristic crossover
	Mutation	Gaussian mutation
EDA: UMDA [18]	Selection	Truncation selection
	Estimation of probabilistic model	$f_g^B(X) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(X-\mu)^2}{2\sigma^2}}$ where $X = \{\lambda_1, \lambda_2, \dots, \lambda_N\}$ , $\mu = \frac{1}{N} \sum_{i=1}^N \lambda_i$ , and $\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (\lambda_i - \mu)^2}$
	Sampling individuals from the model	Sampling normal random variables (individuals) from $f_g^B(X)$ [18, 19]
GA and EDA	Stopping criteria	Terminate when either (i) $g > G^{\max}$ or (ii) $ f_{best}^x - f_{avg}^x  < \delta^{stop}$

individual in a population is mainly represented by the strategy that it has. For the experimental purpose, we consider the range of strategies for  $\lambda_B$  and  $\lambda_S$  in  $[0, 10]$ .

### 3.2 Fitness Function

To evaluate each individual of a population, the fitness function  $f(x)$  is defined as follows:

$$f(x) = U^x(P_x, T_x) = w_P \times U_P^x(P_x) + w_S \times U_S^x(T_x) \quad (5)$$

In each generation  $g$ , randomly pick one individual from  $D_g^B$  and randomly pick the corresponding individual from  $D_g^S$ . Each selection procedure is carried out without a replacement. Then, the selected individuals of one population will negotiate with the selected individuals of the other population in a one-to-one manner. The values of the fitness function will be computed using the resulting negotiation outcomes. Finally, if an agent  $x$  reaches an agreement with its opponent,  $U_P^x(P_x) > 0$ ,  $U_S^x(T_x) > 0$ , and  $f(x) = U^x(P_x, T_x) > 0$ . If a negotiation is terminated without an agreement,  $f(x) = U^x(P_x, T_x) = 0$ .

### 3.3 EAs for Coevolution

The characterizations of GA and EDA used in this work are described in Table 1. For more detailed information of the GA and EDA, refer to [17].

The coevolution procedure for finding the two types of effective strategies (i.e.,  $\lambda_B$  and  $\lambda_S$ ) from the two populations is described in Algorithm 1. The interaction

between the two populations is in the fitness evaluation stage (at lines 3 and 8 in Algorithm 1).

**Algorithm 1** EAs (GA and EDA) for coevolving negotiation strategies

- 1: set the search space  $(\lambda_{\min}, \lambda_{\max})$  and the generation counter  $g \leftarrow 0$
- 2: generate  $D_g^x$  of  $M$  randomly generated individuals
- 3: evaluate individuals in  $D_g^x$  using the fitness function
- 4: **WHILE** the stopping criteria are not satisfied
- 5:     select  $N(\leq M)$  individuals  $d_g^x$  from  $D_g^x$  using the corresponding selection method
- 6:     compute the average fitness values of individuals in  $d_g^x$
- 7:     **GA:** produce  $N$  individuals using the corresponding reproduction mechanism;  
           the  $MP$  consists of the produced  $N$  individuals and  $d_g^x$
- EDA:** produce  $M$  individuals using the corresponding reproduction mechanism;  
           the  $MP$  consists of the produced  $M$  individuals
- 8:     evaluate individuals in  $D_g^x$  using the fitness function
- 9:      $g \leftarrow g + 1$
- 10: **ENDWHILE**
- 11: extract the individual with the highest gene value in high fitness value region

## 4 Problem of P-S-Optimizing Negotiation and Its Solution

### 4.1 Problem of P-S-Optimizing Negotiation

The GA and EDA described in the previous section cannot coevolve the appropriate strategies for *P-S-Optimizing* negotiation because the original fitness function (which is the same as the utility function described in Sect. 3.2) cannot characterize different emphases (i.e., preference levels) on the cost and time criteria for some cases. That is, for the given values of  $w_P$  and  $w_S$ , the utility function  $U^x(P_x, T_x)$  given in Sect. 3.2 cannot appropriately characterize the difference between  $U_P^x(P_x)$  and  $U_T^x(T_x)$  in some cases. For example, consider the case of the exact *P-S-Optimizing* negotiation setting  $(w_P, w_S) = (0.5, 0.5)$ . The total utility  $U^x(P_x, T_x)$  of the set  $U_P^x(P_x) = 0.7$  and  $U_T^x(T_x) = 0.3$  (which corresponds to a more *P-Optimizing* solution) will have the same value for the total utility  $U^x(P_x, T_x)$  of the set  $U_P^x(P_x) = 0.3$  and  $U_T^x(T_x) = 0.7$  (which corresponds to a more *S-Optimizing* solution). Furthermore, it is the same for the set  $U_P^x(P_x) = 0.5$  and  $U_T^x(T_x) = 0.5$  (which corresponds to the exact *P-S-Optimizing* solution). The above example shows that in some cases, the original fitness function cannot characterize the difference between *P-Optimizing*, *S-Optimizing*, and *P-S-Optimizing* solutions.

## 4.2 Proposed Fitness Function

The ambiguity of the original fitness function in the total utility space is solved by adopting the proportion of weighting factors (which represent the preference levels in the total utility space) in the calculation of the proposed fitness function. The key idea of the proposed utility function is to directly measure the difference (or similarity) between (i) the ratio of price and time weighting factors, and (ii) the corresponding ratio of the price and time utility functions. As the difference between them decreases, the value of the fitness function will be considerably high. The proposed fitness function is designed as follows:

$$\text{(Mode 1) for } P\text{-Optimizing: } f(x) = 1 - \left| \frac{w_S}{w_P} - \frac{U_S^x(TC)}{U_P^x(PC)} \right|$$

$$\text{(Mode 2) for } S\text{-Optimizing: } f(x) = 1 - \left| \frac{w_P}{w_S} - \frac{U_P^x(PC)}{U_S^x(TC)} \right|$$

(Mode 3) for *P-S-Optimizing*:

$$f(x) = 1 - \frac{1}{2} \times \left( \left| \frac{w_S}{w_P} - \frac{U_S^x(TC)}{U_P^x(PC)} \right| + \left| \frac{w_P}{w_S} - \frac{U_P^x(PC)}{U_S^x(TC)} \right| \right)$$

## 5 Empirical Evaluation

A series of experiments was carried out to evaluate the performance of the original fitness function (in Sect. 3.2) and the proposed fitness function (in Sect. 4.2).

### 5.1 Experimental Settings

An empirical comparison of GA and EDA is also presented to determine which model is more suitable for coevolving negotiation strategies with different preference criteria for optimizing price and speed. The experimental parameter settings are as in Table 2.

For the purpose of the experiments, both competitive agents have the same weights of the preference criteria for the three negotiation modes. The settings for each negotiation mode are as follows:  $(w_P, w_S) = (1.0, 0.0)$  for *P-Optimizing* negotiation,  $(w_P, w_S) = (0.1, 0.9)$  for *S-Optimizing* negotiation, and  $(w_P, w_S) = (0.5, 0.5)$  for *P-S-Optimizing* negotiation.

### 5.2 Optimal Conditions

In the case of the *P-Optimizing* negotiation, we will experimentally prove the properness by examining two extreme cases: Case I and Case II.



**Table 2** Parameter settings for EAs and negotiation

	Parameters	Values
GA and EDA	Population size ( $N$ )	25
	Mating pool size ( $M$ )	50
	Maximum number of generations ( $G^{\max}$ )	2500
	Threshold for stopping criteria ( $\delta^{\text{stop}}$ )	$10^{-4}$
	Tournament size ( $k$ )	3
	Crossover rate ( $P_{CX}$ )	0.7
	Mutation rate ( $P_{MU}$ )	0.002
Negotiation	$(IP_B, RP_B)$	(5, 85)
	$(IP_S, RP_S)$	(95, 15)
	$(\lambda_{\min}, \lambda_{\max})$	(0, 10)
	Deadline: Long	100
	Deadline: Mid	50

**Case I** If an agent  $B$  has a sufficient bargaining advantage over  $S$  (i.e.,  $\tau_B \gg \tau_S$ ),  $B$  will dominate the negotiation irrespective of whether  $S$  adopts any strategies. The optimal negotiation strategy of  $B$  will be determined by Theorem 1. For example, under the negotiation parameter settings described in Table 1, the optimal strategy of  $B$  is when  $\lambda_B = 3$ , and the agreement is reached at  $P_C = 15$  and  $T_C = 50$ . The strategy of  $S$  and  $\lambda_S$  will not converge to a specific value, and thus, it will have a dynamic range of values. We describe this dynamic range as [*Min. value*, *Max. value*].

**Case II** When both  $B$  and  $S$  do not have any bargaining advantage (i.e.,  $\tau_B = \tau_S$ ), we can think that the agreement of the negotiation with the abovementioned negotiation parameters should be reached at a Pareto optimal point (i.e.,  $P_C = 50$ ). However, in a practical negotiation model, the point does not always follow the Pareto optimal point since the agent that proposes the first negotiation proposal (in our experiment  $S$  does) usually gets a lower payoff (or utility) in the alternating offers protocol. Under the Pareto optimal condition, the optimum strategies are when both  $\lambda_B$  and  $\lambda_S$  equal  $\lambda_{\max} = 10$  (i.e., both  $B$  and  $S$  do not concede at all).

In the case of *S-Optimizing* and *P-S-Optimizing* negotiations, there is no such theory as that in the case of *P-Optimizing* negotiation to prove the optimality of the solutions. However, the evaluation can be carried out by examining whether the solutions follow the general characteristics for the given negotiation mode. For example, in the case of the *S-Optimizing* negotiation, the agreement should be reached at an earlier negotiation time (than its deadline) by paying a relatively high price. The abovementioned two extreme cases are also used for evaluating the optimality of the solutions. Interestingly, according to Proposition 7 given in [13], irrespective of the deadline, agents with linear strategies are more likely to make deals than those

with conservative strategies while achieving higher utility than those with conciliatory strategies. Hence, linear strategies (i.e.,  $\lambda_B = 1.0$  and  $\lambda_S = 1.0$ ) are optimal solutions for the exact *P-S-optimizing* negotiation.

### 5.3 Experimental Results

The results of these two extreme cases for the original utility function and the proposed utility function are shown in Tables 3–6.

#### 5.3.1 Results Obtained Using Original Fitness Function

The results are listed in Tables 3 and 4. The original fitness function in Eq. (5) is the same as the total utility function in Eq. (1).

**Observation 1** Both GA and EDA with the fitness function in Eq. (5) find good strategies for *P-Optimizing* and *S-Optimizing* negotiation, respectively.

**Analysis** In both Case I (in Table 3) and Case II (in Table 4),  $\lambda_B$  and  $\lambda_S$  has close to optimum solutions, as discussed in Sect. 5.1.

**Observation 2** Both GA and EDA with the fitness function in Eq. (5) cannot find effective candidate solutions for the *P-S-Optimizing* negotiation.

**Analysis** In both Case I (in Table 3) and Case II (in Table 4),  $\lambda_B$  and  $\lambda_S$  did not converge to the values that we expected (Similarly, by using the EDA proposed in [5], we found that  $\lambda_B$  and  $\lambda_S$  did not converge in the case of *P-S-Optimization*). This is because more conceding strategies will be cut off as the evolution proceeds since these strategies will get a lower payoff than the other strategies (i.e., linear or conservative strategies) eventually. Hence,  $\lambda_B$  and  $\lambda_S$  of both GA and EDA tend to be more *P-Optimizing*.

#### 5.3.2 Results Obtained Using Proposed Fitness Function

The results are listed in Tables 5 and 6. The proposed fitness function does not equal the utility function any longer. The proposed fitness function measures the similarity between the ratio of weighting factors and the ratio of price and speed utilities. For example, for the *P-Optimizing* negotiation (i.e.,  $(w_P, w_S) = (1.0, 0.0)$ ), and in Case I (i.e., *B* has significant bargaining advantage than *S*), the optimum utilities are obtained at  $(P_C, T_C) = (15, 50)$ . For *B*,  $U_P^B(P_C) = 0.8875$  and  $U_S^B(T_C) = 0.55$ . Hence, the optimum value of the proposed fitness function is  $f_{opt}^B(x) = 1 - \left| \frac{w_S}{w_P} - \frac{U_S^B(T_C)}{U_P^B(P_C)} \right| = 1 - \left| \frac{0.0}{1.0} - \frac{0.55}{0.8875} \right| = 0.3803$ . For *S*,  $U_P^B(P_C) = 0.1$

**Table 3** Results obtained using original fitness function for Case I

<i>(Long, Mid)</i>	GA	EDA
$(w_P, w_S)$	(1.0, 0.0)	(1.0, 0.0)
$\lambda_B$	2.9574	2.9403
$\lambda_S$	[0.1596, 9.8913]	[2.6923, 9.8340]
$(P_C^B, T_C^B)$	(0.5, 0.5)	(0.5, 0.5)
$(P_C^S, T_C^S)$	[0.4624, 2.9979]	2.9524
$(f_{best}^B, f_{avg}^B)$	[0.1012, 9.7045]	[1.3272, 9.4068]
$(f_{best}^S, f_{avg}^S)$	[(15, 30.5074], [8, 50]]	(15, 50)
$N_{Gen}$	(15, 50)	(15.0183, 50)
	(15, 50)	(15.0199, 50)
	(0.8875, 0.8875)	(0.8873, 0.8873)
	(0.1, 0.1)	(0.1002, 0.1002)
	22.06	15.48
	91.8	15.2
	(0.1, 0.9)	(0.1, 0.9)
	0.1110	0.1178
	0.1674	0.1478
	(51.9763, 1.16)	(49.6131, 1.06)
	(52.3486, 1.16)	(49.8052, 1.06)
	(0.9377, 0.9378)	(0.9412, 0.9412)
	(0.9332, 0.9332)	(0.9320, 0.9320)
	63.24	44.98

**Table 4** Results obtained using original fitness function for Case II

<i>(Mid, Mid)</i>	GA	EDA
$(w_P, w_S)$	(1.0, 0.0)	(1.0, 0.0)
$\lambda_B$	9.9931	9.9901
$\lambda_S$	9.7487	9.4845
$(P_C^B, T_C^B)$	(48.1077, 48)	(48.1156, 48)
$(P_C^S, T_C^S)$	(48.1304, 48)	(48.1310, 48)
$(f_{best}^B, f_{avg}^B)$	(0.5150, 0.5150)	(0.5149, 0.5149)
$(f_{best}^S, f_{avg}^S)$	(0.4727, 0.4727)	(0.4727, 0.4727)
$N_{Gen}$	19.16	24.08
	(0.5, 0.5)	(0.5, 0.5)
	9.5089	8.6693
	9.3096	9.9228
	(47.7413, 46.54)	(50.3378, 47.36)
	(47.8476, 46.52)	(50.3659, 47.36)
	(0.3407, 0.3407)	(0.3187, 0.3187)
	(0.3161, 0.3161)	(0.3226, 0.3227)
	35.4	52.48
	67.1	53.16
	(0.9257, 0.9257)	(0.9297, 0.9298)
	(0.9381, 0.9381)	(0.9367, 0.9368)
	(0.1, 0.9)	(0.1, 0.9)
	0.1162	0.1179
	0.1884	0.1659
	(55.2985, 1.1)	(53.0005, 1)
	(55.8283, 1.1)	(53.1861, 1)

**Table 5** Results obtained using proposed fitness function for Case I

<i>(Long, Mid)</i>	GA	EDA
$(w_P, w_S)$	(1.0, 0.0)	(1.0, 0.0)
$\lambda_B$	2.9668	2.9560
$\lambda_S$	[0.0177, 9.9634]	[2.0291, 9.2553]
$(P_C^B, T_C^B)$	(15, 50)	(15.0006, 50)
$(P_C^S, T_C^S)$	(15.0000, 50)	(14.8506, 49.5)
$(f_{best}^B, f_{avg}^B)$	(0.3803, 0.3803)	(0.3803, 0.3803)
$(f_{best}^S, f_{avg}^S)$	(0.0000, 0.0000)	(0.0000, 0.0000)
$N_{Gen}$	134.1789	43.94
	2059.01	18
	(0.5, 0.5)	(0.1, 0.9)
	0.9738	0.1125
	0.9781	0.0025
	(35.3628, 37.96)	(15.8938, 1)
	(35.4753, 37.01)	(15.7906, 1)
	(0.9986, 0.9999)	(0.2256, 0.2257)
	(0.9832, 0.9894)	(0.9998, 0.9998)
	(0.5, 0.5)	(0.1, 0.9)
	0.9847	0.9847
	0.9584	0.9584
	(35.0521, 37.77)	(35.0521, 37.77)
	(35.0540, 37)	(35.0540, 37)
	(0.9941, 0.9941)	(0.9941, 0.9941)
	(0.9745, 0.9746)	(0.9745, 0.9746)
	59.03	17.84
	(0.1, 0.9)	(0.1, 0.9)
	[3.5351, 9.9581]	[3.5351, 9.9581]
	[0.0400, 9.5237]	[0.0400, 9.5237]
	(0, 0)	(0, 0)
	(0, 0)	(0, 0)
	(0, 0)	(0, 0)
	(0, 0)	(0, 0)

Table 6 Results obtained using proposed fitness function for Case II

$(Mid, Mid)$	GA	EDA
$(w_P, w_S)$	(1.0, 0.0)	(1.0, 0.0)
$\lambda_B$	10.0000	9.9919
$\lambda_S$	9.6512	9.4136
$(P_C^B, T_C^B)$	(48.0893, 48)	(48.1109, 48)
$(P_C^S, T_C^S)$	(48.0996, 48)	(48.1432, 48)
$(f_{best}^B, f_{avg}^B)$	(0.7365, 0.7365)	(0.7359, 0.7359)
$(f_{best}^S, f_{avg}^S)$	(0.7121, 0.7121)	(0.7123, 0.7124)
$N_{Gen}$	11.23	17.27
	(0.5, 0.5)	(0.5, 0.5)
	0.9519	0.9508
	1	1
	(50.1952, 28)	(50.1958, 28)
	(50.2000, 28)	(50.2000, 28)
	(0.9909, 0.9910)	(0.9909, 0.9910)
	(1.0000, 1.0000)	(1.0000, 1.0000)
	24.58	19.12
	(0.1, 0.9)	(0.1, 0.9)
	0.1561	0.2595
	0.0026	0.0090
	(15.8203, 1)	(17.4493, 1)
	(15.8099, 1)	(17.4414, 1)
	(0.2167, 0.2167)	(0.2354, 0.2354)
	(1.0000, 1.0000)	(0.9813, 0.9813)
	15.72	41.41

and  $U_S^B(T_C) = 0.1$ . Hence, the optimum value of the proposed fitness function is  $f_{opt}^S(x) = 1 - \left| \frac{w_S}{w_P} - \frac{U_S^x(T_C)}{U_P^x(P_C)} \right| = 1 - \left| \frac{0.0}{1.0} - \frac{0.1}{0.1} \right| = 0.0$ . Likewise, for different preference criteria and negotiation parameter settings, different optimum (i.e., maximum) values of the fitness function will be drawn from the proposed fitness function (e.g., for the above example, the maximum value  $f_{opt}^B(x)$  for  $B$  is 0.3803). More analyses and experiments related to this issue will be presented in our future paper.

**Observation 3** Both GA and EDA with the proposed fitness function find effective strategies for the *P-Optimizing* negotiation.

**Analysis** The results show that both GA and EDA using the new fitness function achieved good performance. In Case I (in Table 5),  $\lambda_B$  converges close to the values of the optimum solutions. Furthermore, in Case II (in Table 6),  $\lambda_B$  and  $\lambda_S$  converge close to the values of the optimum solutions.

**Observation 4** In Case I, the GA with the proposed fitness function finds effective candidate solutions for the *S-Optimizing* negotiation. However, the EDA cannot coevolve *S-Optimizing* strategies. In Case II, both GA and EDA can coevolve effective *S-Optimizing* strategies.

**Analysis** In Case I (in Table 5), the GA can get more conceding strategies that can reach early agreements. However, the EDA cannot find solutions at all. This result shows that the EDA does not have a sufficient search capability to find solutions for the *S-Optimizing* negotiation. In Case II (in Table 6), both GA and EDA can find effective strategies that are considerably more conceding strategies than linear strategies (i.e.,  $\lambda_B = 0.1561$  and  $\lambda_B = 0.0026$  for the GA, and  $\lambda_B = 0.2595$  and  $\lambda_B = 0.0090$  for the EDA).

**Observation 5** In Case I, both GA and EDA with the proposed fitness function find effective candidate solutions for the *P-S-Optimizing* negotiation. The EDA has better performance than the GA in terms of the evolution speed. In Case II, both GA and EDA can coevolve effective *P-S-Optimizing* strategies.

**Analysis** In Case I (in Table 5), the GA needs considerably more generations to converge to solutions than the EDA (i.e., 2059.01 generations for the GA, and 59.03 generations for the EDA). This result shows that the EDA has better performance with respect to coevolving effective *P-S-Optimizing* strategies in terms of the evolution speed (i.e., the number of generations required for the coevolution). In Case II (in Table 6), both GA and EDA can coevolve effective *P-S-Optimizing* strategies that are close to the value of the linear strategy (i.e.,  $\lambda_B = 0.9519$  and  $\lambda_B = 1$  for the GA, and  $\lambda_B = 0.9508$  and  $\lambda_B = 1$  for the EDA).

## 6 Conclusions and Future Work

This paper provides empirical evidence for coevolving negotiation strategies for the negotiation between the two competitive agents having different preference criteria for optimizing price and negotiation speed. Two rather different evolutionary approaches, EDA and GA, were respectively used and compared for coevolving the strategies. The experimental results (given in Tables 3 and 4) showed that adopting the GA and EDA were a good choice for coevolving effective strategies for the *P-Optimizing* and the *S-Optimizing* negotiations. However, we found that both GA and EDA did not converge to proper solutions (that we expected in Sect. 5.1) in the case of the *P-S-Optimizing* negotiation. The main problem of the failure in coevolving the strategies was that the original fitness function (given in [5] and [17]) could not effectively characterize the different weights on the cost and time preference criteria in the total utility space. On the basis of the analysis, we proposed the new utility function in Sect. 4.2. The experimental results (given in Tables 5 and 6) showed that the proposed method achieved more reliable results in the case of the *P-S-Optimizing* negotiation than the method used in [5] and [17]. However, the results also showed that the EDA using the new fitness function did not coevolve effective strategies for the *S-Optimizing* negotiation, and the GA using the new fitness function was not effective in the case of the *P-S-Optimizing* negotiation since it required considerably more generations than EDA. To develop more reliable models that are successful and efficient in all cases, a hybrid model [20] can be adopted to compensate for the defects of each evolutionary approach by combining the GA and the EDA.

Our future work includes an exhaustive analysis that considers (i) more combinations of different preference weights between the cost and the time criteria (e.g., more *P-Optimizing* such as  $(w_P, w_S) = (0.7, 0.3)$  and more *S-Optimizing* such as  $(w_P, w_S) = (0.3, 0.7)$  cases) and (ii) heterogeneous negotiation (e.g., the case when one agent is *P-Optimizing*, while the other agent is *S-Optimizing*, and all these types of cases). Furthermore, by reducing and focusing on the feasible solution space, the restriction scheme of the solution space in [17] can help to reduce the computational overhead and to obtain more high-quality solutions.

**Acknowledgement** This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MEST) (KRF-2009-220-D00092).

## References

1. Rosenschein, J.S., Zlotkin, G.: Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers. MIT Press, Cambridge (1994)
2. Foster, I., Kesselman, C.: The Grid 2: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publishers Inc., San Mateo (2003)
3. Krauter, K., Buyya, R., Maheswaran, M.: A taxonomy and survey of grid resource management systems for distributed computing. *Softw. Pract. Exp.* **32**, 135–164 (2002)



4. Jarek, N., Jennifer, M.S., Jan, W. Eds.: Multicriteria aspects of grid resource management, in: *Grid Resource Management – State of the Art and Future Trends*, pp. 271–295. Kluwer Academic Publishers, Dordrecht (2003)
5. Sim, K.M.: An evolutionary approach for p-s-optimizing negotiation. In: *Evolutionary Computation, 2008, CEC 2008*, pp. 1364–1371. IEEE World Congress on Computational Intelligence (2008)
6. Sim, K.M.: From market-driven agents to market-oriented grids (position paper). *SIGecom Exch.* **5**, 45–53 (2004)
7. Sim, K.M.: Grid commerce, market-driven G-Negotiation, and Grid Resource Management. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **36**, 1381–1394 (2006)
8. Sim, K.M.: A survey of bargaining models for grid resource allocation. *SIGecom Exch.* **5**, 22–32 (2006)
9. Sim, K.M.: Grid resource negotiation: Survey and new directions. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **40**, 245–257 (2010)
10. Sim, K.M., Guo, Y., Shi, B.: Adaptive bargaining agents that negotiate optimally and rapidly. In: *IEEE Congress on Evolutionary Computation, 2007, CEC 2007*, pp. 1007–1014 (2007)
11. Sim, K.M., Guo, Y., Shi, B.: BLGAN: Bayesian learning and genetic algorithm for supporting negotiation with incomplete information. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **39**, 198–211 (2009)
12. Sim, K.M., Choi, C.Y.: Agents that react to changing market situations. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **33**, 188–201 (2003)
13. Sim, K.M.: A market driven model for designing negotiation agents. *Comput. Intell.* **18**, 618–637 (2002)
14. Rubinstein, A.: A bargaining model with incomplete information about time preferences. *Econometrica* **53**, 1151–1172 (1985)
15. Herrera, F., Lozano, M., Verdegay, J.L.: Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. *Artif. Intell. Rev.* **12**, 265–319 (1998)
16. Deb, K., Sindhya, K., Okabe, T.: Self-adaptive simulated binary crossover for real-parameter optimization. In: *The Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, London, England (2007)*
17. Gwak, J., Sim, K.M.: Co-evolving best response strategies for p-s-optimizing negotiation using evolutionary algorithms. In: *Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010, Hong Kong*, pp. 13–18 (2010)
18. Larrañaga, P., Etxeberria, R., Lozano, J.A., Peña, J.M.: Optimization by learning and simulation of Bayesian and Gaussian networks. Technical Report, EHU-KZAA-IK-4/99, University of the Basque Country (1999)
19. Marsaglia, G., Tsang, W.W.: The ziggurat method for generating random variables. *J. Stat. Softw.* **5**, 1–7 (2000)
20. Peña, J.M., Larrañaga, P., Herves, V., Rosales, F., Pérez, M.S.: GA-EDA: Hybrid evolutionary algorithm using genetic and estimation of distribution algorithms, vol. 3029, pp. 1611–3349 (2004)