Proceedings of the International MultiConference of Engineers and Computer Scientists 2010 Vol I,
IMECS 2010, March 17 - 19, 2010, Hong Kong

Co-evolving Best Response Strategies for P-S-Optimizing Negotiation using Evolutionary Algorithms

Jeonghwan Gwak and Kwang Mong Sim*

Abstract—**This paper presents the comparative study of two different evolutionary approaches, a genetic algorithm (GA) and an estimation of distribution algorithm (EDA), in co-evolving negotiation strategies with different preference criteria such as optimizing price and optimizing negotiation speed. Empirical studies demonstrate that both GA and EDA are successful in finding good solutions in price optimizing and speed optimizing negotiation, respectively. However, both are not successful in price and speed concurrent optimizing (P-S-Optimizing) negotiation. From these results, finally, this paper suggests a novel method to find best response strategies for P-S-Optimizing negotiation.**

Index Terms—**Software agent, price and negotiation speed optimizing negotiation, genetic algorithms, estimation of distribution algorithms.**

I. Introduction

In multi-agent systems, negotiation is a process whereby agents exchange messages, make concessions to reduce their differences in the hope of eventually reaching agreements. In other words, agents negotiate to coordinate their activities and to come to mutually acceptable agreements about the division of labor and resources [2, 15].

Sim [4, 6, 8, 16] argued that negotiation agents can play an essential role in realizing the Grid vision. However, there are considerably fewer works on adopting negotiation mechanisms for Grid resource management. Grid resource management involves multiple criteria optimization, and some of these criteria are generally classified into time criteria and cost criteria [5]. It is noted in [1,5] that to maintain good performance of the system, negotiation agents for Grid resource management should be designed to not only optimize (price) utility but should also be successful in negotiation and reach early agreements. This is because any delay in successfully negotiating and acquiring the necessary Grid resources before the deadline for executing a job will be perceived as an overhead. Different consumers may have different preferences [1]. For example, a 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) consumer prefers to obtain a resource more rapidly perhaps by paying a higher price at an earlier round of negotiation. Different emphasis on optimizing price and optimizing negotiation speed can be modeled by placing different weights between them. Exact P-S-Optimization has equal emphasis on price and negotiation speed.

To solve negotiation problems, Sim proposed the *BLGAN* model [9, 17] to support negotiation with incomplete information, and EDA based co-evolutionary model [1] for co-evolving negotiation strategies with different price and negotiation speed preference criteria. This paper is based on [1] and extends the P-S-Optimizing negotiation to the proposed method. Whereas the idea of adopting an EDA for co-evolving negotiation strategies of agents that have different preference criteria such as optimizing price and optimizing negotiation speed is first proposed in [1], this paper provides empirical evidence for comparing EDA and GA in co-evolving negotiation strategies of agents with different preference criteria in optimizing price and negotiation speed.

The rest of the paper is organized as follows. The next section specifies the negotiation model of this work. Section III presents co-evolutionary algorithms using GA and EDA for P-S-Optimizing negotiation. Section IV reports the experiments that are carried out. After that, the problem of the P-S-Optimization and its solution will be presented in section V. Finally, Section VI concludes this work and suggests future directions.

II. Price and Negotiation Speed optimization

As formulated in [1], in 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 reserve prices of *x*. Let D be the event in which *x* fails to reach an agreement with its opponent $\hat{x}$. $U^x : [IP_x, RP_x] \cup D \to [0,1]$ such that $U^x(D) = 0$, and for any possible proposal $P_x \in [IP_x, RP_x]$, $U^x(P_x) > U^x(D)$. If *x* and $\hat{x}$ are sensitive to time, then let τ_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* is within the interval $[RP_S, RP_B]$.

In the bargaining model with complete information between *B* and *S*, both agents know its opponent's initial price, reserve price, and deadline. If one of the agents has significantly longer deadline than its opponent, the agent which has 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 [9, 17] proved that an agent's optimal strategy can be computed using its opponent's deadline and reserve price. It is formulated as follows:

Manuscript received January 12, 2010. This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MEST) (KRF-2009-220-D00092).

Jeonghwan Gwak is with the Department of Information and Communications (DIC), Gwangju Institute of Science and Technology, South Korea (phone: 82-62-715-2280; fax: 82-62-715-2204; e-mail: gwak@ gist.ac.kr).

*Kwang Mong Sim is with the Department of Information and Communications (DIC), Gwangju Institute of Science and Technology, South Korea (phone: 82-62-715-2235; fax: 82-62-715-2204; e-mail: prof_sim_2002@ yahoo.com) (Corresponding Author).

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Theorem [9, 17]. If agent x's deadline τ_x is significantly longer than its opponent's deadline $\tau_{\hat{x}}$, agent x achieves its maximal utility when it adopts the strategy λ_x:

$$\lambda_x = \log_{\frac{\tau_{\hat{x}}}{\tau_x}} \left(\frac{RP_{\hat{x}} - IP_x}{RP_x - IP_x} \right) \quad (1)$$

A. Utility Functions

In this work, the utility function U^X of agent x is defined as in [1].

$$U^X(P_c, T_c) = w_P \times U_P^x(P_c) + w_S \times U_S^x(T_c) \quad (2)$$

where P_c and T_c is the price and time (number of negotiation rounds) at which reaches an agreement. $U_P^x(P_c) \in [0,1]$ 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 weighting factors of B and S respectively, and $w_P + w_S = 1$.

Following [1], the price and speed utilities are given as follows:

$$U_P^x(P_c) = u_{\min}^P + (1 - u_{\min}^P)\left(\frac{RP_x - P_c}{RP_x - IP_x} \right) \quad (3)$$

$$U_S^x(T_c) = u_{\min}^S + (1 - u_{\min}^S)\left(\frac{\tau_x - T_c}{\tau_x} \right) \quad (4)$$

where $u_{\min}^P$ is the minimum utility that x receives a deal at its reserve price, 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 defined as 0.1.

If x does not reach a consensus before its deadline, $U_P^x = U_S^x = 0$, and thus $U^X = 0$. Otherwise, $U^X(P_c, T_c) > 0$.

B. 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 and we adopt time-dependent strategies in [3]. The proposal P_t^x of x at time t, $0 \le t \le \tau_x$ is defined as follows:

$$P_t^x = IP_x + (-1)^\alpha \left(\frac{t}{\tau_x} \right)^{\lambda_x} (RP_x - IP_x), \quad (5)$$

where $\alpha = 0$ for B and $\alpha = 1$ for S are used, and $0 \le \lambda_x \le \infty$.

The time-dependent strategies can be classified into three categories: a) *conservative* (conceding slowly, $\lambda_x > 1$), b) *linear* (conceding linearly, $\lambda_x = 1$), and c) *conciliatory* (conceding rapidly, $0 < \lambda_x < 1$) [3, 18].

C. Negotiation Protocol

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

D. The Objective

For the given different deadlines and different preferences of cost and time criteria (i.e. different w_P and w_S), agents will face different opponents with different deadlines and different strategies. Under these conditions, the objective of this work is to find the best response strategy λ_x that would optimize $U^X(P_c, T_c)$. In this work, learning is based on two asymmetric populations having different fitness evaluations. Agents learn best response strategies by interacting individuals from the other population through random paring. In the following section, the detailed procedure will be described.

III. An Co-Evolutionary GA and EDA For P-S-Optimization

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

In our bilateral negotiation problem domain, inter-population co-evolution having two populations is considered. The fitness of each individual of one population depends on each individual of the other population, and thus an individual's fitness is not fixed but coupled. Therefore, co-evolution is regarded as a kind of landscape coupling where adaptive moves by one individual can potentially change the landscape of the other. In our problem domain, the interaction comes from pairing of strategies between B and S, and thus the successful pairing mechanism is important. To achieve better performance, the resulting pairing should make a sufficiently prevailing set in the feasible set.

In this work, co-evolutionary algorithms using real-coded GA and EDA are implemented to co-evolve best response strategies under different deadlines and different weights of time and speed preferences. B and S have each of their population D^B and D^S consisting of a set of candidate strategies. Evolving the strategies of one population affects the other. In the process of co-evolution, each individual of two populations will negotiate with each other through one-to-one random pairing. The fitness of each individual is determined by its negotiation outcome. Details of the GA and EDA are as follows.

A. Encoding Scheme

Binary coding mechanism has drawbacks due to the existence of Hamming cliffs and the lack of computation precision [13, 14]. Thus, in both GA and EDA, each negotiation strategy of the populations is encoded as real number using a real-coded vector representation. Individuals in D^B and D^S represent each agent's strategies λ_B and λ_S. For the experimental purpose, we consider the range of strategies λ_B and λ_S in [0.1, 10].

B. Fitness Function

The goal of a fitness function is to evaluate each individual in population. In co-evolving the strategies of B and S, the fitness values are determined by the result of negotiation. In

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

the GA and EDA, the fitness function $f(x)$ is defined as follows:

$$f(x)=U^X\left(P_c,T_c\right)=w_P\times U_P^X\left(P_c\right)+w_S\times U_S^X\left(T_c\right) \qquad (6)$$

In each generation *g*, randomly pick one individual from D_g^B, and also randomly pick the corresponding individual from D_g^S. Each selection procedure is done without replacement. The selected individuals will negotiate with each other and fitness function values will be computed using the negotiation outcome. For example, if agents reach an agreement, $U_P^X\left(P_c\right)>0$, $U_S^X\left(T_c\right)>0$, and finally, $f(x)=U^X\left(P_c,T_c\right)>0$. If negotiation is terminated without an agreement, $f(x)=U^X=0$.

In the negotiation process, the optimal consensus, in a sense, for both sides should be made. The strategies that only adapted to one population may not survive because these candidates may result in low fitness values under competitive co-evolution.

C. Selection Process

The purpose of selection is to emphasize the better individuals, also called parents, in candidate solutions and to retain them to be used in the following reproduction stage.

The well-known elitism selection is the method where a limited number of individuals with the highest fitness values are chosen to pass to the next generation without reproduction process. However, in the competitive co-evolution, if the individuals are highly fitted to its own agent's objective, then as the algorithm is executed for more generations they will dominate the whole solution space rapidly, and finally, these may result in poor negotiation outcomes. For this reason, the method is not adopted.

The GA uses *k*-tournament selection which works as follows: *k* individuals are randomly chosen from population. The individual with the highest fitness among *k* individuals is copied to the mating pool (MP). This process is repeated *N* times where *N* is the size of MP. In the *k*-tournament selection, selection pressure is easily controlled by changing the tournament size *k* in that if the k is larger (smaller) strong individuals have a higher (lower) chance to be selected.

The EDA uses the truncation selection as used in [1]. In the truncation selection, individuals are sorted according to their fitness. Then, the top *N* individuals are selected to be saved in MP and to be used to estimate the probability distribution.

D. Reproduction Process

In the GA, there are breeding operators such as crossover (CX) and mutation (MU). From the selected parents, crossover combines them to create offsprings, and mutation enables a parent to create an offspring. There are many existing real-coded crossover methods, for example, intermediate CX, arithmetic CX, heuristic CX, blend CX, and simplex CX. The genetic representation using real-coded genes offers more possibilities for defining mutation. Several forms have been proposed such as Gaussian MU, Cauchy MU, mirror MU, percentage MU, edge MU and tension vector MU. In this work, heuristic CX and Gaussian MU are used.

In the EDA, this process corresponds to the estimation of the probability distribution and sampling solutions from the distribution. In g^{th} generation *N* individuals are selected to form d_g^x. Then, using the *N* individuals μ and σ are estimated using maximum likelihood estimation and is determined by the following formulas:

$$\mu=\frac{1}{N}\sum_{i=1}^{N}\lambda_i \text{, and } \sigma=\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\lambda_i-\mu)^2}$$

The probability distribution is estimated using the following Gaussian distribution:

$$f_g^B(X)=\frac{1}{\sqrt{2\pi\sigma}}e^{-\frac{(X-\mu)^2}{2\sigma^2}} \text{ where } X=\{\lambda_1,\lambda_2,...,\lambda_N\} \qquad (7)$$

Probabilistic logic sampling (PLS) has been the choice in EDAs, but there are other methods which incorporate Gibbs sampling (GS) and find the most probable configurations (MPC). In this work, PLS are used.

E. Stopping Criteria

The co-evolutionary GA and EDA algorithms are terminated when either the maximum number of generations ($G^{\max}$) is reached, or the absolute difference between the average fitness and the best fitness for both D_g^B and D_g^S is smaller than 1e-4.

1. Set the search space $(\lambda_{\min},\lambda_{\max})$ and generation counter $g=0$.
2. Generate each populations D_g^B of *M* uniformly distributed individuals.
   Generate each populations D_g^S of *M* uniformly distributed individuals.
3. Evaluate individuals in populations D_g^B and D_g^S using fitness function.
4. While the stopping criteria are not met do
   a. Select a subset d_g^B of the population D_g^B using selection method.
      Select a subset d_g^S of the population D_g^S using selection method.
   b. Compute the average fitness in d_g^B and d_g^S.
   c. Apply crossover to d_g^B.
      Apply crossover to d_g^S.
   d. Apply mutation to d_g^B.
      Apply mutation to d_g^S.
   e. Evaluate individuals in d_g^B and d_g^S using fitness function.
   f. Create new population with d_g^B and D_g^B.
      Create new population with d_g^S and D_g^S.
   g. g = g+1
5. Return the best individual with the best fitness and the highest genes from D_g^B and D_g^S, respectively.

Fig. 1. The GA for co-evolving strategies of P-S-optimizing Negotiation

1. Set the search space $(\lambda_{\min},\lambda_{\max})$ and generation counter $g=0$.
2. Generate each populations D_g^B of *M* uniformly distributed individuals.
   Generate each populations D_g^S of *M* uniformly distributed individuals.
3. Evaluate individuals in populations D_g^B and D_g^S using fitness function.
4. While the stopping criteria are not met do
   a. Select a subset d_g^B of the population D_g^B using selection method.
      Select a subset d_g^S of the population D_g^S using selection method.
   b. Compute the average fitness in d_g^B and d_g^S.
   c. Estimate the probability distribution $f_g^B(X)$ of *X* from d_g^B.
      Estimate the probability distribution $f_g^S(Y)$ of *Y* from d_g^S.
   d. Sample *M* individuals from d_g^B and d_g^S respectively
   e. Evaluate individuals in d_g^B and d_g^S using fitness function.
   f. g = g+1
5. Return the individual with the best fitness and the highest gene value from D_g^B and D_g^S, respectively.

Fig. 2. The EDA for co-evolving strategies of P-S-optimizing Negotiation

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

IV. Empirical Evaluation

The performance evaluation for finding successful negotiation strategies of the P-S-Optimizing negotiation will be compared using the co-evolutionary algorithms with the GA and EDA described in Section III. Each experiment was repeated 500 times using the same condition. If the values of each experiment converge to a specific value, then the resulting values in the resulting table represents the mean value. Otherwise, in the case of dynamic range, we represent it as [minimum value, maximum value] in the table.

A. Experiment settings

For the experimental purpose, the experimental parameter settings are as follows:

TABLE 1
The Parameter Settings for the GA and EDA

Parameters	Values
Population size (M)	25
Mating pool size (N)	50
Maximum number of generations ($G^{\max}$)	500
Crossover rate (P_{CX})	0.7
Mutation rate (P_{MU})	0.002

TABLE 2
The Parameter Settings for the Negotiation

Parameters	Values
(IP_B, RP_B)	(5, 85)
(IP_S, RP_S)	(95, 15)
$(\lambda_{\min}, \lambda_{\max})$	(0.1, 10)
(*Long, Mid*)	$(\tau_B = 100, \tau_S = 50)$
(*Long, Long*)	$(\tau_B = 100, \tau_S = 100)$
(*Mid, Mid*)	$(\tau_B = 50, \tau_S = 50)$
(*Short, Short*)	$(\tau_B = 25, \tau_S = 25)$

B. Optimal conditions (only for the P-Optimizing case)

When an agent B has sufficient bargaining advantage over S, in our experiment (*Long, Mid*), then the negotiation outcome follows the theorem in equation (1) as described in section II. In this case, B will dominate the negotiation regardless of the strategy of S. The value of λ_B with the above negotiation parameters is computed as follows:

$$\lambda_B = \log_{\frac{\tau_S}{\tau_B}}\left(\frac{RP_S - IP_B}{RP_B - IP_B}\right) = \frac{\log\left(\frac{RP_S - IP_B}{RP_B - IP_B}\right)}{\log\frac{\tau_S}{\tau_B}} = \frac{\log\left(\frac{95-5}{85-5}\right)}{\log\frac{50}{100}} = 3$$

According to the theorem, when B adopts the strategy λ_B = 3, the agreement is made at P_c = 15 and T_c = 50, and the strategy of S, λ_S, will not converge to a specific value and thus it will have dynamic range of values.

When both B and S do not have bargaining advantage, the agreement of the negotiation with the above negotiation parameters should be made at the Pareto optimal point, i.e. $P_c = 50$.

The results of these two extreme cases are shown in Tables 4 and 5 respectively. Also, in the experiments three distinctive negotiation types were used: P-Optimizing, (w_P, w_S) = (1.0, 0.0), S-Optimizing, (w_P, w_S) = (0.1, 0.9) and P-S-Optimizing, (w_P, w_S) = (0.5, 0.5), negotiation.

C. The results of P-Optimizing Negotiation

When B has sufficient bargaining advantage, both GA and EDA found the values of λ_B which are close to the optimum value, 2.9574 in the GA and 2.9403 in the EDA. On the other hand, any values of strategies of S are possible in making contract, that is, the values of λ_S have dynamic range in the GA and EDA, [0.1596, 9.8913] in the GA and [2.6923, 9.8340] in the EDA. The agreement point of the GA is at P_c=15 and T_c= 50. In the case of the EDA, the agreement point is at T_c=50 but P_c was made at a slightly higher value than the optimal value. Details are given in Table 4.

When both has no bargaining advantage, both B and S reach an agreement at $P_c \approx 48.1$, the optimum is at $P_c = 50$, and $T_c = 48$ in both GA and EDA, respectively.

Accordingly, we can conclude that both GA and EDA find good candidate solutions for P-Optimizing negotiation.

D. The results of S-Optimizing Negotiation

When B has sufficient bargaining advantage, both B and S finish the negotiation near T_c=1 and the agreement price P_c is near the Pareto optimal point. The reason is that because the negotiation finishes rapidly, the effect of price negotiation is of no great importance.

The case without bargaining advantage also has a value near T_c =1. However, both have the agreement price P_c at somewhat higher values than the Pareto optimal point, P_c=50. Using the GA, B obtained P_c =55.2985 and S obtained P_c=55.8283. Using the EDA, B obtained P_c= 53.0005 and S obtained P_c= 53.1861.

In this case, both GA and EDA find good candidate solutions for S-Optimization but the EDA has slightly better performance in finding the optimum values.

E. The results of P-S-Optimizing Negotiation

When B has sufficient bargaining advantage, the GA obtained a range of values of P_c= [15, 30.5074] and T_c= [8, 50]. Hence, P_c and T_c did not converge. In the case of the EDA, P_c and T_c converge to near the optimal values of P-Optimizing negotiation, which is not the values of P-S-Optimizing negotiation that we have expected.

Under the condition that there is no bargaining advantage, the GA and EDA find P_c near the Pareto optimal point close to the deadline, which is also not what we have expected.

In this type of negotiation, for both GA and EDA, λ_B and λ_S did not converge to the values that we have expected (Similarly, using the EDA in [1], λ_B and λ_S did not converge for the case of P-S-optimization).

V. The Problem of P-S-Optmization And Its Novel Solution

A. The Problems of P-S-Optimizing Negotiation

The objective of P-Optimizing negotiation is to find the best strategies which maximize the utility function (3). Also, the objective of S-Optimizing negotiation is to find the best strategies which maximize the utility function (4). After fitness evaluation, the population, D_g^x, of an agent x is

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

consisted of weighted sum of the two objectives. However, as negotiation rounds proceeds, the P-S-Optimizing solution candidates which are more S-optimizing, in which the candidates make urgent agreement, will be severely affected by the number of negotiation rounds. Therefore, the candidates will be discarded rapidly as negotiation rounds increased. For these reasons, the negotiation results are inclined to be more P-optimizing as described in section IV. *E*. In addition, GA and EDA with different parameter settings, especially, large population size, may converge to a proper solution but it will take extremely large number of generations, and thus it is not realistic in practice.

B. The Proposed Methods

To solve this problem, the novel P-S-optimizing method, using restriction scheme of feasible solution (strategy) space is suggested in this section. The idea comes from the characteristics of the range of negotiation strategies [3, 18] (see section II. *B*). In response to different deadlines, an agent adopts different strategies in making concessions [18]. With a longer deadline, an agent may find it advantageous to adopt more a conservative strategy since it has plenty of time for negotiating deals (Proposition 5, [18]). A conciliatory strategy may be more suitable if an agent is coerced to complete a deal rapidly (Proposition 6, [18]). Regardless of deadline, agents with linear strategies are more likely to make deals than with conservative strategies while achieving higher utility than with conciliatory strategies (Proposition 7 [18]). Hence, to finish negotiation in an earlier time than its deadline, an agent has to select conciliatory strategies, $0 < \lambda_x < 1$, at the expense of price. In this case, both price and negotiation speed is able to be adjusted simultaneously.

Following propositions 5-7, we define three different negotiation speed options (S^{mode}): *fast*, *moderate* and *slow*. The range of strategy λ, search space in the GA and EDA, is restricted to the given range in each mode. By choosing one of the modes, we can select the degree of price and negotiation speed. The settings for the proposed method are in the following Table 3. For the experiments, the same experimental parameters in Tables 1 and 2 were used.

TABLE 3
The Parameter Settings for the Proposed Method

Negotiation Speed Mode (S^{mode})	The Range of Strategy
Fast	$\lambda_{\min} \le \lambda < \lambda_1$
Moderate	$\lambda_1 \le \lambda < \lambda_2$
Slow	$\lambda_2 < \lambda \le \lambda_{\max}$

C. Experimental Results

To verify the correctness of the proposed method, we consider two extreme cases again: (a) *B* has significant bargaining advantage. And (b) both *B* and *S* have no bargaining advantages. In these experiments, $\lambda_1 = 0.5$ and $\lambda_2 = 1.0$ were used for the purpose of experiment. The results are in Tables 7 and 8.

As you can see, the proposed method is able to do P-Optimizing, S-Optimizing and P-S-Optimizing negotiation moderately. In the *slow* mode, it is able to find a strategy that reaches near optimal price close to the deadline and it almost reaches the theoretical optimum value. In the *fast* mode, it can reach an agreement at an earlier time and thus, in this sense, it is able find an S-Optimizing negotiation strategy. Finally, in the *moderate* mode, it can find a P-S-Optimizing negotiation strategy that reaches an agreement near a moderate point in terms of price and speed. The same analysis used in section IV can be applied to analyze the proposed method.

VI. Conclusions and Future Work

In this work, we have studied two kinds of co-evolutionary algorithms using the GA and EDA in price and speed optimizing negotiation problem domain to find the best response strategies.

The experimental results show that adopting the GA and EDA are an effective choice to co-evolve best response strategies for P-Optimizing and S-Optimizing negotiation. However, we found that both GA and EDA do not converge to proper values for P-S-Optimizing negotiation.

From the analysis, we proposed the novel method using restriction scheme of feasible solution space to solve the difficulties in P-S-Optimization. And we showed that the proposed method is appropriate in optimizing price and negotiation speed simultaneously. Due to the space limitation this paper only reports the preliminary results on the issue. Extensive empirical results on the proposed P-S-Optimization case will be reported in a future paper. And more complex negotiation model [6, 10] will be considered.

For future study, GA-EDA hybrid model [11] or other possible candidates, for example, real-coded Bayesian optimization algorithm [12], can be adopted in the co-evolution process. In addition, random pairing issue discussed briefly in Section III, and adjusting selection pressure are also important issues to solve the difficulties of P-S-Optimizing negotiation as mentioned in the section V. *A.* and to expect further improvements.

References

[1] K. M. Sim, "An evolutionary approach for p-s-optimizing negotiation," in *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, 2008, pp. 1364-1371.
[2] J. Rosenschein and G. Zlotkin, Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers. Cambridge , MA : MIT Press, 1994.
[3] K. M. Sim and C.Y. Choi. Agents that React to Changing Market Situations. IEEE Transaction on Systems, Man and Cybernetics, Part B: Cybernetics, Vol. 33, No. 2, pp 188-201, April 2003.
[4] K. M. Sim, “From market-driven agents to market-oriented Grids,” *ACM SIGECOM: E-Commerce Exchange*, vol. 5, no. 2, pp. 45–53, Nov. 2004.
[5] K. Kurowski et al. “Grid Resource Management – State of the Art and Future Trends”, chapter “Multicriteria Aspects of Grid Resource Management”, pages 271-295, Kluwer Academic Publishers, 2003.
[6] K. M. Sim, G-Commerce, Market-driven G-Negotiation Agents and Grid Resource Management. IEEE transactions on Systems, Man and Cybernetics, Part B, 2006, Vol. 36, No. 6, Dec. 2006, pp. 1381-1394.
[7] A. Rubinstein, “Perfect equilibrium in a bargaining model,” Econometrica, vol. 50, no. 1, pp. 97-109, 1982.
[8] K. M. Sim. Grid Resource Negotiation: Survey, Analysis and New Directions. To appear in IEEE Transactions on Systems, Man and Cybernetics Part C (2010)
[9] K. M. Sim, *et al.*, "Adaptive bargaining agents that negotiate optimally and rapidly," in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, 2007, pp. 1007-1014.
[10] L. Cuihong, *et al.*, "Bilateral negotiation decisions with uncertain dynamic outside options," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 36, pp. 31-44, 2006.
[11] J.M. Peña, P. Larrañaga, V. Herves, F. Rosales and M.S. Pérez, "GA-EDA: Hybrid Evolutionary Algorithm Using Genetic and Estimation of Distribution Algorithms," vol. 3029, pp. 1611-3349, 2004.
[12] C. W. Ahn and R. R. S., "On the Scalability of Real-Coded Bayesian Optimization Algorithm," *Evolutionary Computation, IEEE Transactions on*, vol. 12, pp. 307-322, 2008.
[13] F. Herrera, *et al.*, "Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis," *Artif. Intell. Rev.*, vol. 12, pp. 265-319, 1998
[14] K. Deb, *et al.*, "Self-adaptive simulated binary crossover for real-parameter optimization," presented at the Proceedings of the 9th annual conference on Genetic and evolutionary computation, London, England, 2007.

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

[15] M. Wooldridge, *An Introduction to MultiAgent Systems*: John Wiley & Sons, 2009.
[16] K. M. Sim. A survey of bargaining models for grid resource allocation. ACM SIGECOM: E-commerce Exchanges Vol. 5(No. 5), pp. 22–32 (January, 2006)
[17] K. M. Sim, *et al.*, "*BLGAN*: Bayesian Learning and Genetic Algorithm for Supporting Negotiation With Incomplete Information," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 39, pp. 198-211, 2009.
[18] K. M. Sim, "A Market Driven Model for Designing Negotiation agents," *Computational Intelligence*, vol. 18, pp. 618-637, 2002.

TABLE 4
The Results of Co-evolving Strategies using the GA and EDA with sufficient Bargaining Advantage

(*Long*, *Mid*)	GA			EDA		
(w_P, w_S)	(1.0, 0.0)	(0.5, 0.5)	(0.1, 0.9)	(1.0, 0.0)	(0.5, 0.5)	(0.1, 0.9)
λ_B	2.9574	[0.4624,2.9979]	9.9785	2.9403	2.9524	0.1178
λ_S	[0.1596, 9.8913]	[0.1012, 9.7045]	9.6486	[2.6923, 9.8340]	[1.3272, 9.4068]	0.1478
(P_c^B, T_c^B)	(15, 50)	([15, 30.5074], [8,50])	(51.9763, 1.16)	(15.0183, 50)	(15, 50)	(49.6131, 1.06)
(P_c^S, T_c^S)	(15, 50)	([15, 30.5074], [8,50])	(52.3486, 1.16)	(15.0199, 50)	(15.0003, 50)	(49.8052, 1.06)
(f_{best}^B, f_{avg}^B)	(0.8875, 0.8875)	(0.7429, 0.7429)	(0.9377, 0.9378)	(0.8873, 0.8873)	(0.7188, 0.7188)	(0.9412, 0.9412)
(f_{best}^S, f_{avg}^S)	(0.1, 0.1)	(0.2025, 0.2025)	(0.9332, 0.9332)	(0.1002, 0.1002)	(0.1000, 0.1000)	(0.9320, 0.9320)
N^{Gen}	22.06	91.8	63.24	15.48	15.2	44.98
(*Long*, *Mid*) represents the scenario when $\tau_B = 100$ and $\tau_S = 50$. In this case *B* has a significant bargaining advantage in terms of time over *S*						

TABLE 5
The Results of Co-evolving Strategies using the GA and EDA with no Bargaining Advantage

(*Mid*, *Mid*)	GA			EDA		
(w_P, w_S)	(1.0, 0.0)	(0.5, 0.5)	(0.1, 0.9)	(1.0, 0.0)	(0.5, 0.5)	(0.1, 0.9)
λ_B	9.9931	9.5089	0.1162	9.9901	8.6693	0.1179
λ_S	9.7487	9.3096	0.1884	9.4845	9.9228	0.1659
(P_c^B, T_c^B)	(48.1077, 48)	(47.7413, 46.54)	(55.2985, 1.1)	(48.1156, 48)	(50.3378, 47.36)	(53.0005, 1)
(P_c^S, T_c^S)	(48.1304, 48)	(47.8476, 46.52)	(55.8283, 1.1)	(48.1310, 48)	(50.3659, 47.36)	(53.1861, 1)
(f_{best}^B, f_{avg}^B)	(0.5150, 0.5150)	(0.3407, 0.3407)	(0.9257, 0.9257)	(0.5149, 0.5149)	(0.3187, 0.3187)	(0.9297, 0.9298)
(f_{best}^S, f_{avg}^S)	(0.4727, 0.4727)	(0.3161, 0.3161)	(0.9381, 0.9381)	(0.4727, 0.4727)	(0.3226, 0.3227)	(0.9367, 0.9368)
N^{Gen}	19.16	35.4	67.1	24.08	52.48	53.16
(*Mid*, *Mid*) represents the scenario when $\tau_B = 50$ and $\tau_S = 50$. In this case both *B* and *S* has no bargaining advantage in terms of time						

TABLE 6
The Proposed Method of Co-evolving Strategies using the GA and EDA with sufficient Bargaining Advantage

(*Long*, *Mid*)	GA			EDA		
Speed mode	slow	moderate	fast	slow	moderate	fast
λ_B	2.9615	0.9999	0.4992	2.9901	0.9998	0.4998
λ_S	[0.1215, 9.9984]	0.9884	0.4975	[0.1274, 9.8615]	0.9732	0.4998
(P_c^B, T_c^B)	(15, 50)	(34.5778, 37.94)	(41.8578, 21.98)	(15, 50)	(34.6067, 37.98)	(41.9069, 22)
(P_c^S, T_c^S)	(15, 50)	(34.6269, 38)	(41.8963, 22)	(15, 50)	(34.6190, 38)	(41.9259, 22)
(f_{best}^B, f_{avg}^B)	(0.8875, 0.8875)	(0.6672, 0.6672)	(0.5853, 0.5853)	(0.8875, 0.8875)	(0.6669, 0.6669)	(0.5847, 0.5848)
(f_{best}^S, f_{avg}^S)	(0.1004, 0.1004)	(0.3208, 0.3208)	(0.4026, 0.4026)	(0.1, 0.1)	(0.3207, 0.3207)	(0.4028, 0.4029)
N^{Gen}	22.06	23.24	21.94	19	23.24	33.66
(*Long*, *Mid*) represents the scenario when $\tau_B = 100$ and $\tau_S = 50$. In this case *B* has a significant bargaining advantage in terms of time over *S*						

TABLE 7
The Proposed Method of Co-evolving using the GA and EDA with no Bargaining Advantage

(*Long*, *Long*)	GA			EDA		
Speed mode	slow	moderate	fast	slow	moderate	fast
λ_B	9.9785	0.9960	0.4996	9.9886	0.9999	0.4996
λ_S	9.6486	0.9961	0.4997	9.4727	0.9959	0.4997
(P_c^B, T_c^B)	(48.1470, 95)	(49.8699, 56.82)	(49.6986, 31.96)	(48.1297, 94.98)	(49.8069, 56.96)	(49.7152, 32)
(P_c^S, T_c^S)	(48.1663, 95)	(49.8917, 56.86)	(49.7331, 31.96)	(48.1334, 95)	(49.8256, 57)	(49.7342, 32)
(f_{best}^B, f_{avg}^B)	(0.5146, 0.5146)	(0.4952, 0.4952)	(0.4971, 0.4971)	(0.5149, 0.5149)	(0.4959, 0.4959)	(0.4969, 0.4969)
(f_{best}^S, f_{avg}^S)	(0.4731, 0.4731)	(0.4925, 0.4925)	(0.4907, 0.4907)	(0.4727, 0.4728)	(0.4917, 0.4918)	(0.4907, 0.4908)
N^{Gen}	19.72	23.58	22.88	22.42	24.7	30.02
(*Long*, *Long*) represents the scenario when $\tau_B = 100$ and $\tau_S = 100$. In this case both *B* and *S* has no bargaining advantage in terms of time						