ARTICLE IN PRESS

Ain Shams Engineering Journal xxx (xxxx) xxx

Contents lists available at ScienceDirect

Ain Shams Engineering Journal

journal homepage: www.sciencedirect.com

Electrical Engineering

A chaotic owl search algorithm based bilateral negotiation model

Walaa H. El-Ashmawi [a,c,*], Diaa Salama Abd Elminaam [b,c], Ayman M. Nabil [c], Esraa Eldesouky [d,a]

[a] Computer Science Department, Faculty of Computers and Informatics, Suez Canal University, Ismalia City, Egypt
[b] Information Systems Department, Faculty of Computers and Artificial Intelligence, Benha University, Benha City, Egypt
[c] Computer Science Department, Faculty of Computer Science, Misr International University, Cairo, Egypt
[d] Department of Computer Science, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Saudi Arabia

ARTICLE INFO

Article history:
Received 28 October 2019
Revised 23 December 2019
Accepted 2 January 2020
Available online xxxx

Keywords:
Owl search algorithm
Bilateral negotiation
Chaos
Meta-heuristic algorithms
Logistic chaotic map
Negotiation tactics

ABSTRACT

Negotiation is a core activity between stakeholders with different goals. This paper offers a solution based on the automated bilateral negotiation model using a recent meta-heuristic algorithm Owl Search Algorithm (OSA), and chaos theory. The proposed algorithm called Chaotic Owl Search Algorithm (COSA). This algorithm is used to adapt negotiation strategies for computing negotiation offers throughout the negotiation process. For this aim, a negotiation grasped between two parties during several negotiation rounds. The results of the proposed algorithm compared with the standard OSA, PSO and the most common negotiation tactics. Different control parameters considered for accurate judgments of the suggested optimization techniques. The comparative study proved that the COSA provides accurate results over compared algorithms in terms of Average buyers' /seller's utility, Average negotiation rounds and Average processing time. This paper is the first of integrating chaos theory with the OSA in optimization problems and especially in the negotiation process.

1. Introduction

Negotiation is a process among two or more parties (i.e., human or software agents) who are interacting to achieve a mutual agreement over a set of negotiation issues. Negotiation becomes a core activity in human society for any commerce or business. It can be seen as a successful communication way for solving transaction conflicts and produce effective deals between commerce entities [1]. Negotiation plays a prominent part in various domains, including artificial intelligence [2–6], game theory [7,8], commerce systems [9,10], and economics [11].

During the negotiation process, different parties (e.g., one-to-one, one-to-many, or many-to-many) and various issues (e.g., price, quality, lead time, quantity, etc.) can be considered. Price is the most prominent example of negotiation issuees[12]. Even so, quality is considered a vital issue. At a certain level of price, one party requires high-quality commodities or more advanced services, while the other party can only afford low-quality products or fewer services[13]. Another critical issue is the lead time, as products cannot be obtained immediately and require more time to be available. Depending on the number of negotiation parties, a bilateral or multilateral negotiation model is proposed [14-16]. The purest form of negotiation model consists of two parties and a single negotiation issue [17] or multi-issues of bilateral negotiations [18,19]. In real-world negotiations, a multi-issue negotiation is not an easy as a single-issue negotiation [20] due to the extensive negotiation state space, which causes an achievement of a sub-optimal solution rather than an optimal one. In multi-issues negotiation, agreements must be obtained over all the negotiation issues.

Several approached were introduced for automating negotiation, such as game theory, heuristic, and argumentation approaches [21]. The authors in [22] proposed a bilateral negotiation model with formal game theory. This model is based on incomplete information about parties, and negotiation has been done under time constraints. In [23], the authors analyzed the bilateral negotiation process and obtained the outcome under different negotiation scenarios based on the availability of the opponent's' information to the agent — their work based on a negotiation decision functions to reach an agreement. The authors

* Corresponding author.
E-mail addresses: w.hashmawi@ci.suez.edu.eg (W.H. El-Ashmawi), diaa.salama@fci.bu.edu.eg (D.S. Abd Elminaam), ayman.nabil@miuegypt.edu.eg (A.M. Nabil), em.eldesouky@psau.edu.sa (E. Eldesouky).

Peer review under responsibility of Ain Shams University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.asej.2020.01.005
2090-4479/© 2020 Ain Shams University. Production and hosting by Elsevier B.V.

Nomenclature

IP_b	Aspiration price of the buyer
BP_b	Reserved price of the buyer
IP_s	Aspiration price of the seller
BP_s	Reserved price of the seller
Z	Bargaining zone
B	Set of buyers
S	Set of sellers
I	Set of issues
n	Number of buyers
m	Number of sellers
l	Number of issues
Min_i and Max_i	The minimum and maximum acceptable value of issue i, respectively
v_i	The score value of issue i
w_i	The weight of issue i
t	Negotiation round
T_{dead}	Maximum negotiation rounds
$\mathcal{O}$	An offer
$\mathcal{O}_{goal}$	The goal offer
$\mathcal{A}^t$	Negotiation action at round t
$U(\mathcal{O})$	Utility of offer $\mathcal{O}$
SL	Agent satisfaction level
ϑ	The concession rate
o_i	An Owl$i$
o_{ij}	The j^{th} dimension of the i^{th} Owl
o_L and o_U	The lower and upper bounds of i^{th} owl in j^{th} dimension, respectively
In_i	Intensity information of i^{th} owl
$f(o_i)$	The fitness value of i^{th} owl
o_{best}	Global best owl
o_{worst}	Global worst owl
R_i	distance information of i^{th} owl
V	location of prey
Ic_i	Changed intensity of i^{th} owl
p_{vm}	The probability of prey movement
α	Generated random number
β	Exploration constant
o_i^t	The position of i^{th} owl at time t
o_i^{t+1}	The position of i^{th} owl at time $t+1$
P	Population of owls
$C(t)$	The chaotic sequence generated by chaotic map at independent run t
a	Parameter controls the behavior of $C(t)$
$\mathcal{L}$ and $\mathcal{U}$	The lower and upper boundaries of negotiation issues
$\mathcal{O}_{counter}$	Counter-offer
$\mathcal{O}_{accept}$	Accepted offer
$maxp$ and $minp$	The maximum and minimum price, respectively
$maxq$ and $minq$	Maximum and minimum level of allocated service, respectively
Acc_a^t	An acceptable value of negotiation party a at time t
Acc_{high} and Acc_{low}	The highest and lowest acceptance value, respectively
AR	Agreement ratio
Ag_b and Ag_s	The number of successful agreement by the buyer and seller, respectively
Neg_{total}	The total number of negotiation cases
$AvgU_b$ and $AvgU$	Average buyer and seller utility, respectively
Agr_{total}	The total number of negotiation agreement cases
UB_i,US_i	The utility of buyer/seller i
RB_{b_i}	The reserved price of buyer i
IB_{s_i}	The open price of seller i
Acc_s, Acc_b	The accepted seller's / buyer's offer price
$AvgN$	The average negotiation rounds
N_i	The total number of rounds required to reach a mutual agreement at negotiation process i
$AvgT$	Average processing time
T_i	The time of the negation process i to reach acceptance
$AvgF_b$, $AvgF_s$	The average buyer/seller fitness value
$f_{b_i}(O), f_{s_i}(O)$	The buyer's / seller's global best fitness value of the i^{th} negotiation round
f_{COSA}	The best fitness value of the proposed chaotic owl search algorithm
f_{PSO} and f_{OSA}	The best fitness value of the PSO and standard owl search algorithm, respectively

in [24] proposed a negotiation model between buyer and seller over a single-issue. Their model is based on a Markov Decision theory. Each negotiation party has a private negotiation deadline and tried to maximize their outcomes. For multi-issue negotiations, a bilateral negotiation based on graph theory is proposed [25] to model the opponent's preferences. Although a game theory model [26] can be used for analyzing the optimal solutions theoretically under a specific situation such as bilateral negotiation, it suffers from providing an action plan to be followed by negotiation parties for reaching the optimal solution.

On the other side, heuristic approaches [27] tend to explore the search space in a non-exhaustive way in order to produce sound, rather than optimal solutions. The authors in [28] presented a genetic algorithm to model the matching procedure between parties in bilateral negotiation. This approach requires an estimation of the utility function of the opponent for computing the fitness function of each chromosome. While in [29], each chromosome represents a rule of negotiation instead of a negotiation offer. In their work, the computation of fitness function based on the number of agreements to be reached. An enhanced GA is proposed in [30] based on introducing a new operator called "trade." This operator simulates a concession making procedure of the negotiation process. The main drawback of this approach is based on a centralized negotiation mechanism in which the preferences of all negotiation parties should be available to the centralized mechanism. All the above approaches require complete information about negotiation spaces between parties. However, the authors in [31] developed a genetic algorithm-based adaptive negotiation model under a time constraint. They have proven by theoretical the optimality of the negotiation mechanism and opens the door for effective negotiation systems. The authors in [32] presented a demonstration of a software system for one-to-one negotiation. This system incorporates a heuristic approach for guessing the opponent's preferences from the history of his offers. This approach acts as a training tool for human negotiators.

In the classic negotiation methods, all information about negotiation parties should be available. This cannot be true in real-life negotiations because parties are disinclined to reveal their private information such as reservation price, deadline, or their strategies to opponents in order to avoid any exploitation from the other party. This issue can be slightly avoided by extracting some information from the exchanged offers between negotiators to achieve a better agreement. This information can be helpful to learn some aspect of the opponent model. The authors in [33] presented an overall survey of bilateral negotiation concerning existing opponent models. Therefore, making a feasible mutual agreement in the negotiation process under incomplete information about the opponent is a challenging problem. Another solution is to involve a mediator or a broker agent in the negation process for evaluating offers. [34] Proposed a negotiation approach for the e-commerce system through the presence of a mediator agent. The authors in [35] proposed a bilateral negotiation model between two parties over two issues (price and quantity). The mediator agent is responsible for determining whether there is a deal opportunity between two parties or not. In the case of a mediator, negotiators need to expose their preferences to him. Hence, trust becomes a vital problem that is not suitable in real-life situations. the authors in [36] proposed a sealed-bid bilateral negotiation mechanism. Each negotiated party submit his offer to a mediator for evaluation.

Although heuristic approaches can be executed in less time than yield near-optimal results, meta-heuristics are generic structures to design problem-specific heuristic algorithms. Meta-heuristics are widely used for various problems [37]. The authors in [38,39] developed a negotiation model based on simulated annealing (SA) for evaluating the responses of agents by the mediator. At each negotiation round, a single alternative proposal is controlled by the mediator. While in [40], agents should be accepted a certain number of proposals over time. Recent efforts have been made on swarm intelligence algorithms for negotiation in the commerce field. The authors in [41] proposed an improved version of the particle swarm optimization (PSO) algorithm to find an efficient negotiation solution to prevent it from being trapped in local optima. While the authors in [42] increase the efficiency of negotiation and decrease the cost of it in order to match both buyers' and sellers' requirements. Moreover, a bilateral negotiation model based on particle swarm optimization is proposed in [43]. This model for selecting the best player(s) to trade in the electricity market. This model integrated with the decision support system to determine which player's action is the best from all actions. Such integration can add value to find a better solution. In this paper, a recent swarm optimization algorithm is used to find the best offer (counter-offer) among all possible offers for each party in the negotiation process. The authors in [44] attempted to present a negotiation based multi-objective PSO to ease the limitation of computational resources. Their model does not guarantee to reach an optimal solution for all negotiation cases, especially they have tested the model on three buyers and three sellers. The proposed model in this research is tested on various sets of experiments with a different number of buyers and sellers.

Though many meta-heuristic algorithms can find a feasible solution, it can stick to a local optimum. Therefore, meta-heuristic optimization algorithms can improve their search capability by integrating it with chaos theory. Chaos is the most suitable approach due to various characteristics such as dynamic, nonrepetitive, and ergodicity [45]. The dynamics ensure the variety of solutions under different search spaces, and nonrepetitive, and ergodicity enhances the searching speed. It can be widely used in various applications [46]. A comprehensive review of chaos embedded meta-heuristic optimization algorithms discussed in [47]. This review presents a list of chaotic maps, and based on their results, it cannot be easy to determine which chaotic map that performs best. Different chaotic maps are discussed in [48] to improve the performance of PSO by updating the parameters. The same as in [49], it can be integrated with harmony search (HS) for performance improvement.

For the sake of taking the advantages of both meta-heuristic algorithm and chaos theory, in this paper, we have integrated one of the popular chaotic map (chaotic logistic map) into a most recent optimization algorithm (OSA) in the bilateral multi-issue negotiation model for feasible agreement between negotiators. This integration enables each negotiation party to choose the appropriate line for choosing offers at every negotiation round. The proposed model has a twofold: first, our solution does not require a mediator for evaluation offers, being completely decentralized. The second is based on a recent meta-heuristic algorithm integrated with a chaos theory known as COSA-based negotiation. The proposed algorithm evolves a population of available offers for each party in the direction of generating a new offer. The logistic chaotic map embedded owl search optimization algorithm not only converges towards an optimal negotiation solution but also enhances the variety of movement towards available offers.

This paper is organized as follows. The mathematical formulation of the bilateral negotiation model presents in Section 2. Section 3 illustrates the basic structure of the owl search algorithm, while in Section 4, the proposed chaotic OSA-based negotiation model illustrated. The test results and performance analysis of the proposed algorithm against others are shown in Section 5. The main conclusion points of this paper and some future suggestions are listed in Section 6.

2. Bilateral negotiation model

As well known, negotiation is the exchange of relevant offers by both parties to reach a feasible agreement. There are two fundamental aspects that should be highly considered during the design of the negotiation model. One is the negotiation protocol, and the other is the negotiation strategies [50–52]. Negotiation protocol defines the space of possible agreement that each party can achieve along with the action that a negotiating party can make (i.e., rules of the encounter between agents [14]). Negotiation strategy demonstrates the negotiator's behavior during the negotiation process and determines when and how to act (i.e., agents' negotiation behaviors through a set of tactics [51]). The negotiation protocol is known as to be public where parties know the rules of negotiation as well while the strategy is known to be private to each party (i.e., each party has its strategy). Each party in the negotiation process makes proposals defined by its protocol and using its strategy.

In the context of bilateral negotiation, two parties are involved (i.e., buyer and seller) that have contradictory demands and exchange proposals over various issues (e.g., price, quality, quantity, etc.) during a serious of threads (i.e., rounds) to reach a deal. The negotiation process terminates when one of them reaches an agreement or a deadline reached. Therefore, the feasible solution is achieved by finding a deal range between two parties during the negotiation process. A simple bilateral negation process [53] can be seen in Fig. 1. Each party has a private aspiration zone, which is a maximum or minimum range that must be respected in order to reach a deal. The intersection between the parties' aspiration zone is known as a bargaining zone.

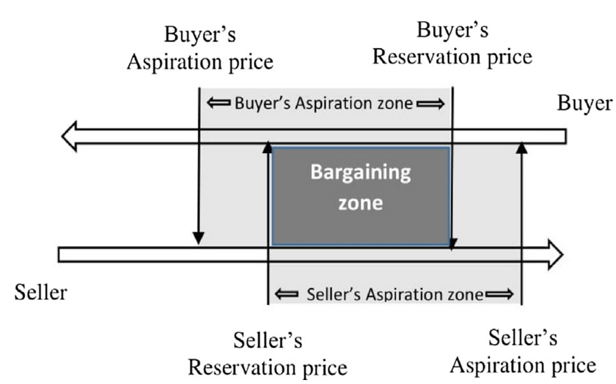


Fig. 1. Simple bilateral bargaining zone.

The aspiration price of the buyer and the reserved price are represented as IP_b and BP_b respectively, while the aspiration price of the seller and the reserved price are represented as IP_s and BP_s Respectively. The bargaining zone Z (i.e.,$Z = [IP_b, BP_b] \cap [IP_s, BP_s]$) is the overlapping region between buyer's and seller's reservation prices. Both parties reach a successful agreement $Z! = \varnothing$.

2.1. Mathematical model of the negotiation process

In the proposed model, we consider a set of n buyers $B = \{b_1, b_2, \cdots, b_n\}$ where $n \geq 1$ and a set of m sellers $S = \{s_1, s_2, \cdots, s_m\}$ where $m \geq 1$, negotiate over a set of l issues $I = \{i_1, \cdots, i_l\}$; $l \geq 1$. Min_i and Max_i are the acceptable range of values for an issue i (i.e., $i \in [Min_i, Max_i]$). For simplicity, each issue has a mapping score value $v_i : [Min_i, Max_i] \to [0, 1]$ to normalize it. Each party has a weight w_i towards an issue i that reflects the importance of that issue.

Negotiation advances in threads $t(t \in R^+)$ where negotiating parties make decisions to make offers $\mathcal{O}$s. At each thread (i.e., round) during the negotiation, the negotiation action $\mathcal{A}^t(a_j \to a_k)$ from negotiating party aj to negotiating party ak at time t can take place. In the case of $a_j = b$, $a_k = s$ or $a_j = s$, $a_k = b$ and $\mathcal{A}^t = \{accept, counter\_offer, reject\}$.

This model follows an alternating offers protocol [54] that is widely used in negotiation models [55]. Fig. 2 illustrated the general structure of the protocol by considering two negotiating parties (buyer b and seller s). The negotiation process terminates if an agreement is reached (b/s accepts the offer) or the negotiation deadline is reached.

From Fig. 2, evaluating the offer means measuring the utility of an offer $\mathcal{O}$ received from the opponent relative to his own goal $\mathcal{O}_{goal}$. The utility of an offer ($\mathcal{O}$) based on normalized issues' values and can be computed according to equation (1) and (2):

$$U(\mathcal{O}) = \sum_{i=1}^{l} w_i * v_i(r_i) \text{ where } \sum_{i=1}^{l} w_i = 1 \tag{1}$$

$$v_i(r_i) = \begin{cases} \frac{max_i - r_i}{max_i - min_i} v_i \text{ decreases as } r_i \text{ increases} \\ \frac{r_i - min_i}{max_i - min_i} v_i \text{ increases as } r_i \text{ increases} \end{cases} \text{where } v_i(r_i) : R^+ \to [0,1] \tag{2}$$

Each party can take its action based on the distance between the utility values of the two offers. For simplicity, the distance can be easily computed as the absolute value of the numerical difference of offers' utility values as in equation (3).

$$V(\mathcal{O}) = |U(\mathcal{O}) - U(\mathcal{O}_{goal})| \tag{3}$$

The closer the distance, the higher to accept the offer. Therefore, the offer is accepted if the value of it is greater than or equal to the agent satisfaction degree SL (i.e., it is a maximum or minimum satisfaction level defined by both buyer and seller). Otherwise, the

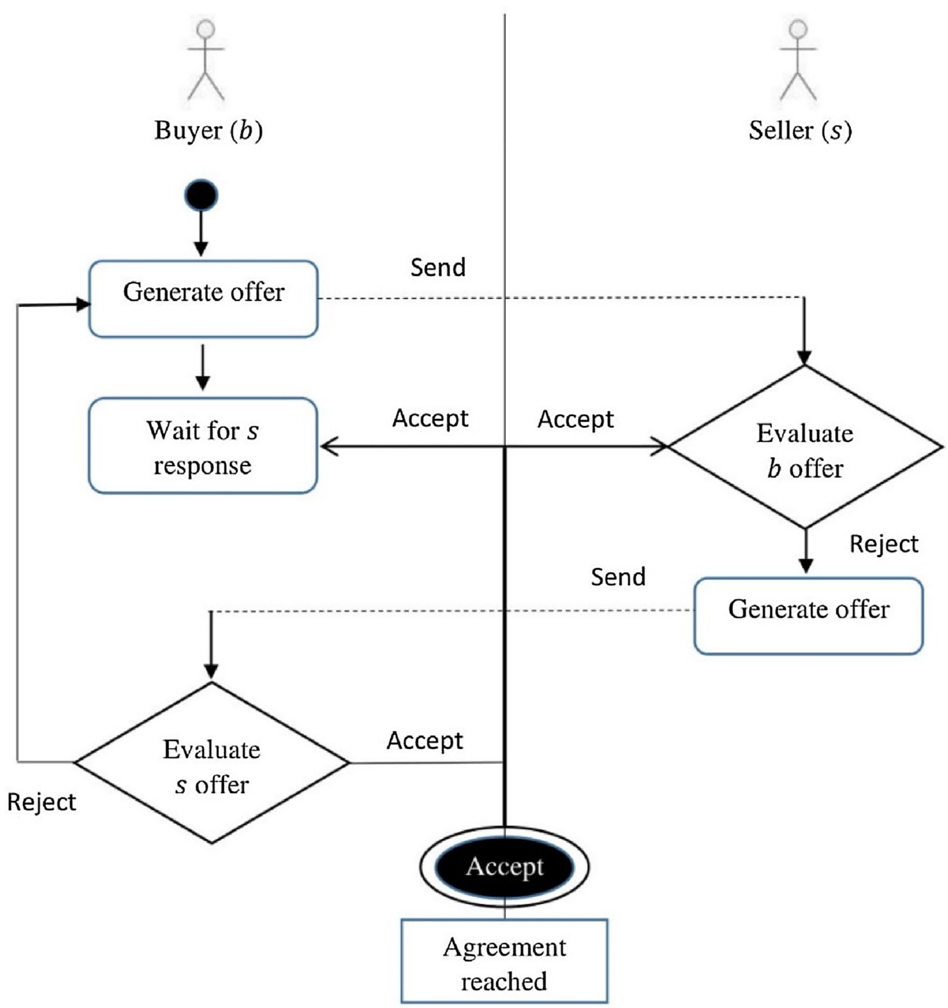


Fig. 2. Structure of alternating-offer protocol.

opponent generates a new offer (counter-offer) and sends it to the other while the deadline not reached. The action at time t can be taken by any party according to equation (4).

$$\mathcal{A}^t = \begin{cases} AcceptifV(\mathcal{O}) \geq SL \\ Exitt > T_{dead}(deadline) \\ Counter - offerotherwise \end{cases} \tag{4}$$

The most common negotiation tactics for generating a counteroffer is time-dependent tactics [56]. This tactic computes the value of a negotiation issue i by considering the time factor T. Equation (5) is used by a_j, which represents either a buyer or seller to generate a new counteroffer for a_k at time t.

$$\mathcal{O}_i^t(a_j \to a_k) = \begin{cases} min_i^j + T(t)\left(max_i^j - min_i^j\right) for \;\; decreasing \;\; v_i^j \\ min_i^j + (1 - T(t))\left(max_i^j - min_i^j\right) for \;\; increasing \;\; v_i^j \end{cases} \tag{5}$$

$$T(t) = (t/T_{dead})^{1/\vartheta} \tag{6}$$

Where max_i^j, min_i^j are the maximum and minimum acceptable range of issue i for agent j.v_i^j is the value of issue i for agent j. ϑ is strategic parameters that determine the concession rate.

3. Owl search algorithm

The Owl Search Algorithm (OSA) is a nature-inspired recent optimization algorithm developed by Mohit Jain et al. (2018)[57]. It simulates the hunting behavior of barn owls that rely on their hearing capability to find prey (vole) in the night rather than sight.

Owls are nocturnal birds that have some main characteristics, such as binocular vision, binaural hearing, and feathers adapted for silent flight [58]. One of the species is barn owl that has a distinct feature of the auditory system with the vertical asymmetry of ears [59,60] which makes the sound reaches one ear before the other. They hunt in total darkness [61] and mainly depend on hearing for finding prey. Additionally, these owls have developed exceptional sound localization abilities.

OSA is a population-based optimization algorithm that consists of an initial population of owls' position in the search space, evaluating the fitness of owls and updating mechanism for new positions of owls. These stages can be summarized below:

Stage 1 (initial population): The initial set of random solutions to an optimization problem represent the initial positions of owls in the forest. The population contains n owls (i.e., individuals), and each one is represented by a d-dimensional vector, as represented below.

$$o_i = (o_{i1}, o_{i2}, \cdots, o_{id}) \tag{7}$$

Where o_{ij} represents the j^{th} dimension of the i^{th} Owl. In order to allocate the initial position of each owl in the forest, equation (8) can be used.

$$o_i = o_L + Rand(0,1) * (o_U - o_L) \tag{8}$$

Where o_L and o_U are the lower and upper bounds of i^{th} owl in j^{th} dimension respectively. $Rand(0,1)$ is a generated random number in the range $[0,1]$.

Stage 2 (Owl's evaluation): each owl's location is evaluated on a specific fitness function f. By considering the fitness, value relates to intensity information received through owl's ear, the best owl o_{best} is assigned to the one that receives max intensity (for maximization problems) and min intensity (for minimization problems). At the same time, the worst owl o_{worst} is assigned to the one that receives min intensity (for maximization problems) and max intensity (for minimization problems). For normalizing the intensity information In of the i^{th}owl by using equation (9).

$$In_i = \frac{f(o_i) - o_{worst}}{o_{best} - o_{worst}} \tag{9}$$

Where $o_{worst} = min(\{f(o_i) : i = 1, \cdots, n\})$ and $o_{best} = max(\{f(o_i) : i = 1, \cdots, n\})$.

Stage 3 (updating owls' location): Each owl computes the distance information R_i to the prey according to equation (10).

$$R_i = \|o_i, V\|_2 \tag{10}$$

Where V is the location of prey that was achieved by fittest owl;$V = o_{best}$ (i.e., the global optimum in the forest). Due to silent flights of owls towards the prey, they receive changed intensity respect with the inverse square law of sound intensity [62] as computed below. *Random noise* for more realistic.

$$Ic_i = \frac{In_i}{R_i^2} + Randomnoise \tag{11}$$

Form the behavior of owls and their ability to fly silently (i.e., changing their positions). Therefore, their movement depends on a probability to allocate a new position, as seen in equation (12).

$$o_i^{t+1} = \begin{cases} o_i^t + \beta * Ic_i * \left|\alpha V - o_i^t\right| p_{vm} < 0.5 \\ o_i^t - \beta * Ic_i * \left|\alpha V - o_i^t\right| p_{vm} \geq 0.5 \end{cases} \tag{12}$$

Where p_{vm} is the probability of prey movement, α is a random number in the range [0,0.5], and β is a constant that is linearly decreasing from 1.9 to 0 through iterations, which permits a substantial exploration of the search space. Algorithm 1 presents the pseudocode of these steps. The owl search algorithm has proven its efficiency and effectiveness in solving global optimization problems.

Algorithm 1. Owl search algorithm (OSA)

```
Set the population size n.
Set t = 0.
Generate randomly o_i^t according to Eq. (8), i = 1, ..., n
Repeat
   for each o_i^t do
         Evaluate f(o_i^t)
   End for each
   Set o_best is the global best solution in the population
   Set o_worst is the worst solution in the population
    For each o_i^t do
         Compute the intensity of owl In_i according to Eq. (9)
         Compute the distance information of each owl R_i
   according to Eq. (10)
         Compute the changed intensity of each owl Ic_i
   according to Eq. (11)
    End for each
    For (i = 0; i < n; i++) do
         set p_vm = rand(0,1)
         Compute o_i^{t+1} according to Eq. (12);
    End for
    Update o_best
    Set t = t + 1.
Until stopping criteria is satisfied
Report the best owl's location o_best
```

4. A chaotic owl search algorithm based negotiation model (COSA)

This section presents the chaotic owl search algorithm (COSA) based negotiation model. For initializing the negotiation process, each negotiation party employs an owl search algorithm to define the initial population (i.e., its proposals) of offers according to its search space. Throughout the negotiation process, a set of concession rounds is consists of alternate placements of offers and counter-offers to be evaluated to decide their acceptability. Based on the last offer received from the opponent, each party updates its population.

4.1. Chaotic owl negotiation mapping schema

In OSA, the population can be seen as a set of all owls in the forest, while in the COSA-based negotiation model, the population consists of a set of possible offers for each negotiation party. Table 1 lists the chaotic owl-based negotiation mapping schema.

Both negotiation parties (i.e., the buyer and the seller) conducted the COSA for computing the offer and determine the best one to be exchanged as a counter-offer during the negotiation process. Each owl in the COSA forest corresponds to an offer $\mathcal{O}$ that contains information about all the issues under consideration (e.g., Offer $\mathcal{O}_1$), as shown in Fig. 3. Precisely, each owl consists of as many as the number of negotiation issues. This number is fixed among all the negotiation parties and the i^{th} negotiation issue is denoted as $Neg\_issue_{(i)}$. For each party, the population P of owls is used to represent a subset of available offers.

An illustrative example of an offer $\mathcal{O}$ with four negotiation issues (price, quality, lead time, and quantity) is shown in Fig. 4. The value for each negotiation issue can be generated within the buyer's/seller's aspiration zone.

Due to the importance of the diversity of the initial population to enable the population to spread in search space, the initial population of the proposed algorithm is generated by a chaotic map. The logistic map is one of the simplest maps; it appears in the nonlinear dynamics of a biological population that evidencing the chaotic behavior [45], which represents mathematically by Eq. (13).

$$C(t+1) = a * C(t) * (1 - C(t)) \tag{13}$$

Where $C(t)$ is the t^{th} chaotic number at each independent run t. a is the driving parameter that controls the behavior of $C(t)$ which equals 4 in the experiments; $a = 4$ and $C(t) \in (0, 1)$. Therefore, in the proposed algorithm COSA, the initial position of owls in the population p can be generated according to the following pseudocode.

Pseudocode 1: generation of initial population$P$

For $i = 1$ to$|P|$
  For $j = 1$ to$I$
  $\mathcal{O}_{ij} = \mathcal{L} + C(t) * (\mathcal{U} - \mathcal{L})$
  End for
$\mathcal{O}_i = (\mathcal{O}_{i1}, \mathcal{O}_{i2}, \ldots, \mathcal{O}_{iI})$
End for

$\mathcal{L}$ and $\mathcal{U}$ are the boundaries of the lower and upper value of negotiation issues, respectively. $C(t)$ is the chaotic sequence generated by chaotic maps. This integration of chaos and owl search algorithm can produce a proper distribution by the characteristic of random and ergodicity of chaos.

Table 1
COSA-based negotiation mapping schema.

OSA	Chaotic Owl-based negotiation
Dimensional search space (d)	Set of negotiation issues (I)
Owl (o)	Possible offer$\mathcal{O}$
Population	Set of possible offers$P$
Inverse intensity	Effect of counter-offer to reach agreement
Evolution of population	Computing new offers
Fittest owl	Counter-offer

4.2. Negotiation protocol

The proposed model is designed for bilateral (i.e., one-to-one) multi-issues, time-dependent functions. It can also be suitable for multi-lateral negotiation by using multiple bilateral (i.e., one-to-many) negotiation. The owl-based negotiation model consists of mainly three stages, as described below. Algorithm 2 illustrated these stages in detail.

Stage 1 (pre-negotiation stage): In this stage, each negotiation party specifies his goal offer $\mathcal{O}_{goal}$. Assigns a weight to each negotiation issue based on his preferences and defines the negotiation characteristics such as the deadline of the negotiation process. Moreover, each party populates its initial population P with owls closer to the specified goal. The global best owl is defined as a goal and remains the best during the negotiation process.
Stage 2 (negotiation stage): The negotiation process follows an alternating offer protocol. It is started by sending the buyer its own goal as the first negotiated offer to the seller. Subsequently, exchanges of offers (e.g., counter-offers $\mathcal{O}_{counter}$) between negotiation parties for moving in the direction of agreement or till a deadline is reached. The main stages of owl search algorithm based negotiation are invoked at each party of the negotiation for determining the best counter-offer as described in algorithm 2.
Stage 3 (negotiation result): At the end of the negotiation process, one action can be taken (accept or reject or deadline reached). If one of the parties accept the offer, $\mathcal{O}_{accept}$ is determined by the two negotiation parties. Any party can reject the offer if its fitness value less than a minimum acceptable value. If the negotiation deadline reached, the negotiation ended.

Fig. 3. Chaotic owl-based negotiation representation.

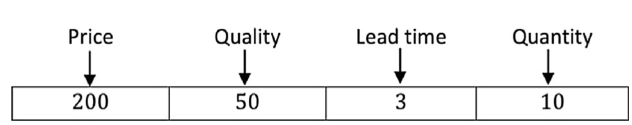


Fig. 4. An illustrative example of offer $\mathcal{O}$.

Algorithm 2. Chaotic Owl-based negotiation model (COSA)

1. Initialize the negotiation process through the pre-negotiation phase
2. Define $\mathcal{O}_{goal}$, P (*initial population*) according to **Pseudocode 1**
3. Set $t = 1$; T_{dead} (deadline of negotiation process)
4. **Send the first offer to the opponent** SendOffer($\mathcal{O}_{goal}$)
5. **While** $(t < T_{dead})$ **do**
6. Evaluate offer $f(\mathcal{O}_{counter})$ according to Eq. (3).
7. Compute Acc_a^t according to Eq. (14).
8. While $(f(\mathcal{O}_{counter}) \ < \ Acc_a^t)$ do /// invoke owl search algorithm
9. Repeat
10. Update population $(P, \mathcal{O}_{counter})$
11. For each $\mathcal{O}_i^t \in P$ do
12. Compute $f(\mathcal{O}_i^t)$;$f(\mathcal{O}_i^t) = |U(\mathcal{O}_i^t) - U(\mathcal{O}_{counter})|$
13. End for each
14. Set $\mathcal{O}_{best}^t = min(\{f(\mathcal{O}_i^t) : i = 1, \ldots, n\})$ // minimization problem
15. Set $\mathcal{O}_{worst}^t = max(\{f(\mathcal{O}_i^t) : i = 1, \ldots, n\})$ // minimization problem
16. For each $\mathcal{O}_i^t \in P$ do // compute intensity and changed intensity
17. Compute $In(\mathcal{O}_i^t)$; $In(\mathcal{O}_i^t) = \frac{f(\mathcal{O}_i^t) - \mathcal{O}_{best}^t}{\mathcal{O}_{worst}^t - \mathcal{O}_{best}^t}$ // intensity
18. Compute $R(\mathcal{O}_i^t)$; $R(\mathcal{O}_i^t) = \ (\mathcal{O}_i^t - \mathcal{O}_{best}^t)$ // distance information
19. Compute $Ic(\mathcal{O}_i^t)$; $Ic(\mathcal{O}_i^t) = \frac{In(\mathcal{O}_i^t)}{R(\mathcal{O}_i^t)^2}$ // changed intensity
20. End for each
21. Set $C_1(t)$ and $C_2(t)$ according to Eq. (16). // integrate the chaotic map
22. For each $\mathcal{O}_i^t \in P$ do
23. Compute $\mathcal{O}_i^{t+1}$;$\mathcal{O}_i^{t+1} = \mathcal{O}_i^t + C_1(t) \ \ * Ic(\mathcal{O}_i^t) * \left|(C_2(t) * \mathcal{O}_{counter}) - \mathcal{O}_i^t\right|$
24. End for each
25. Update$\mathcal{O}_{best}^t$
26. Until termination criteria satisfied
27. $\mathcal{O}_{counter} = \mathcal{O}_{best}^t$
28. Send to opponent SendOffer($\mathcal{O}_{counter}$)
29. currentOffer = WaitForNewOffer()
30. End while
31. If $(f(\mathcal{O}_{counter}) \ < \ Acc_{low})$ then
32. Send to opponent reject offer($\mathcal{O}_{reject}$)
33. Break
34. Else
35. Send to opponent accepted offer($\mathcal{O}_{accept}$)
36. Break
37. End if
38. Set$t = t + 1$
39. **End while**
40. **End of the negotiation process**

Fig. 5 shows the main steps of the negotiation process between two negotiators (buyer b and seller) over two issues (price p and level of allocated service q). In the model, b and s have the opposite goal. The seller tends to request a maximum price (*maxp*) and provide a minimum level of allocated service (*minq*). On the opposite, the buyer will try to pay a minimum price (*minp*) and need a maximum level of allocated service (*maxq*). The buyer starts the negotiation by sending his goal offer $\mathcal{O}_{goal}$ To the seller. Once a new offer is received, the negotiated party evaluates the incoming offer concerning his goal (Line 5). Consequently, the offer is checked whether it can be accepted or not according to an acceptable value of party at the current time Acc_a^t (Line 7). The Acc_a is varied over time. At the beginning of the negotiation process, the accepted value for each party is set to the highest value Acc_{high} Moreover, it decreased over time. In the end, each party will scale its acceptance value towards the specified minimum value Acc_{low}. The Acc_a^t can be computed as follows:

$$Acc_a^t = Acc_{low} + \left(Acc_{high} - Acc_{low}\right) * \left(t/T_{dead}\right)^{\vartheta} \tag{14}$$

where Acc_{high} and Acc_{low} are the highest and lowest acceptance value of the offer (i.e., it is a user-defined based on his preferences), t is the current time and T_{dead} is the deadline of the negotiation process (i.e., T_{dead}can be either defined as an absolute time or maximum number of negation rounds), and ϑ is a concession rate (i.e., a positive non-zero value set by the negotiator before the negotiation starts) that reflects negotiator's attitude toward the agreement. If $\vartheta = 1$, the negotiator is neutral to concession while if $\vartheta > 1$, the negotiator is willing to concede quickly to reach an agreement with other parties. For simplicity, Acc_{high} is set to 1 and $Acc_{low} \in (0, 1)$ for normalized acceptance value.

If the fitness value of the offer $f(\mathcal{O}_{counter})$ is below the computed acceptance value at the current time, the agent samples for new offers from its search space (i.e., current population), and selects the best as a counter-offer to be sent through the stages of Owl (Lines 9–26). Each owl (i.e., offer) in the population evaluated according to a fitness function $f(\mathcal{O}_i^t)$. It demonstrates the most cost-effective offer against the counter-offer as in equation (15). It represents the distance between the utility of the current offer and the utility of the offer received from the opponent. The least the distance, the fittest offer.

$$f(\mathcal{O}_i^t) = \left|U\left(\mathcal{O}_i^t\right) - U(\mathcal{O}_{counter})\right| \tag{15}$$

The fittest offer is assigned to $\mathcal{O}_{best}^t$ while the worst offer is assigned to $\mathcal{O}_{worst}^t$. To enrich the searching behavior and movement of an owl, a chaotic logistic map can be used in this paper. In the proposed chaotic owl search algorithm COSA, the most crucial parameter β which promotes the exploration of search space is replaced by a sequence of chaotic maps. Besides, the random generator α is modified by chaotic maps according to equation (13). The movement of each owl in the COSA to a new location depends on the chaotic movement of the counter offer. Therefore, the updating Eq. (12) can be reformulated as in Eq. (16).

$$\mathcal{O}_i^{t+1} = \mathcal{O}_i^t + C_1(t) \ \ * Ic(\mathcal{O}_i^t) * \left|(C_2(t) * \mathcal{O}_{counter}) - \mathcal{O}_i^t\right| \tag{16}$$

Where $C_1(t)$ and $C_2(t)$ are the chaotic map values based on equation (16) that have a significant influence on the changed intensity of current owl and the opposite offer respectively.

The counter-offer will be the global best (fittest) owl in the population (Line 27) after the counter-offer is being determined by negotiation party, it sends to the opponent (Line 28) and waits for a negotiation action $\mathcal{A}^t = counter\_offer$ (Line 29). Once an offer is received more significant than the defined acceptance value at current negotiation round Acc_a^t, the negotiator sends an acceptance message ($\mathcal{A}^t = accept$) to the opponent (Lines 35–40). If the evaluated offer less than the minimum acceptance Acc_{low} then, the negotiated party can reject the offer and exit the negotiation process ($\mathcal{A}^t = reject$) (Lines 31–40).

The figure illustrated the case where the buyer interacts with a seller to reach an agreement. This interaction can be done through a finite-horizon negotiation under incomplete knowledge of the characteristics of an opponent. The negotiation process involves several alternating offers tied with a negotiation deadline (e.g., the maximum number of rounds), and negotiators tend to maxi-

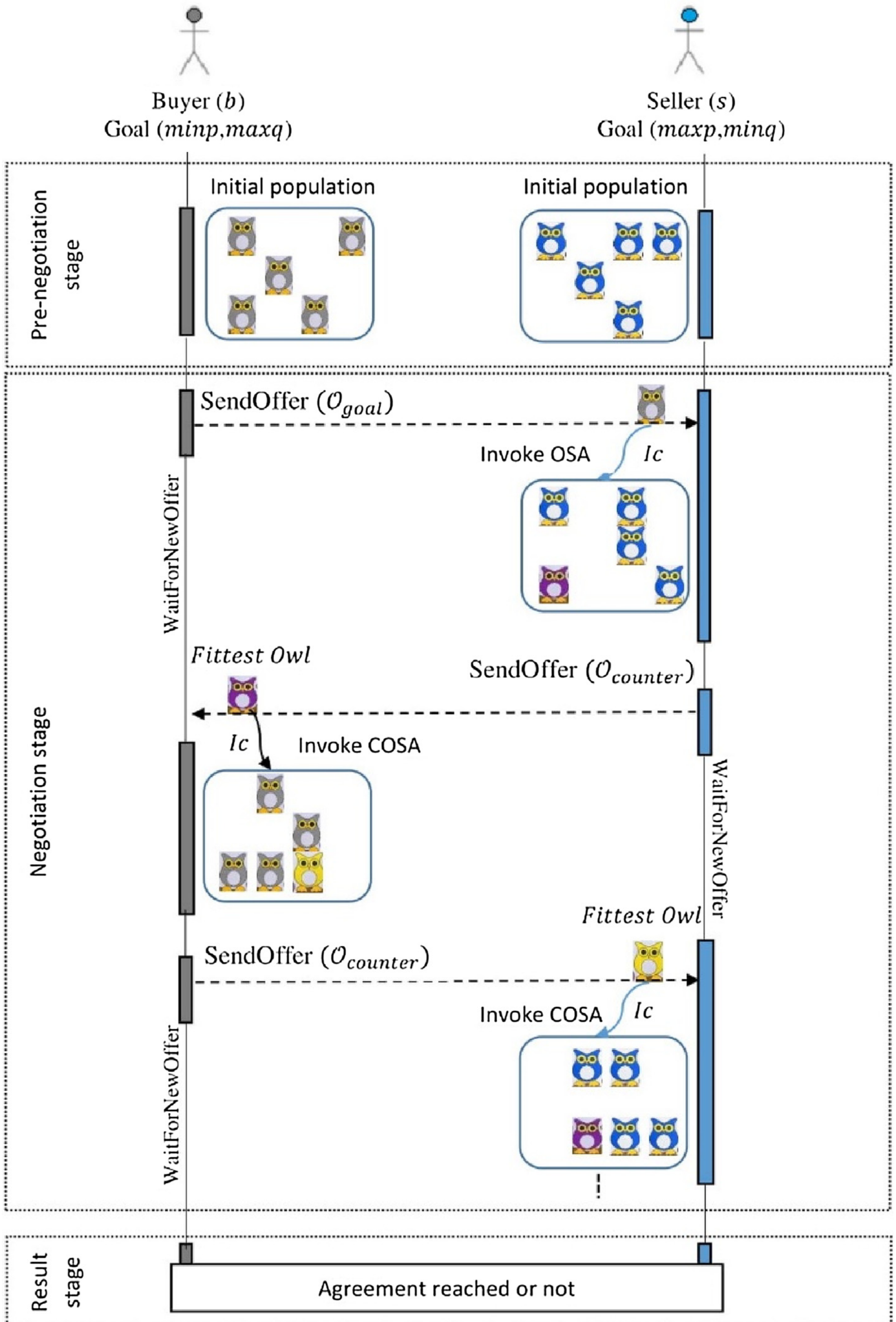


Fig. 5. chaotic Owl-based negotiation process.

mize their own goal while ensuring that an agreement is reached [63].

5. Experimental results and performance analysis

For evaluating the chaotic owl-based bilateral negotiation model, several series of experiments are conducted to demonstrate the quality of the proposed solution in terms of a feasible agreement. Besides, we compared the results with three other existing models to ensure the effectiveness of using a chaotic owl search algorithm as a decision making for forming better offers. The first one is the standard owl search algorithm (OSA) [57]), the second is the most used swarm algorithm (PSO [64]), and the third is the essential tactical decision functions for generating offers (Tact [28]).

Each experiment consists of several buyers, several sellers, and several products that are assigned to sellers randomly. The buyer and seller negotiated over a set of issues (e.g., price and quality) for each product. The parameters for each experiment are listed in Tables 2 and 3. At the beginning of each experiment, each buyer *b* and seller *s* allocated their open and reservation values for negotiation issues uniformly distributed randomly from the acceptable range of negotiation issues [Min_i, Max_i] as in Table 2. The buyer and seller represent a bilateral negotiation situation. All the

Table 2
Experiment parameters' list.

Exp. No.	No. of buyers	No. of sellers	No. of products
1	10	5	3
2	20	10	5
3	30	20	8
4	40	25	10
5	50	30	10

experiments are implemented by using Eclipse Java Neon V-1.8 running on Intel(R) Core i7 CPU-2.80 GHz with 8 GB RAM and operating system (Windows 10).

5.1. Parameter setting

The parameter setting of the compared algorithms is illustrated in this subsection — different experiments with the varying numbers of buyers, sellers, and products as listed in Table 2.

For each negotiation issue, there is an acceptable aspiration zone where the negotiation agreement between buyer and seller is reached. Table 3 listed the acceptable (minimum and maximum) range of the two issues (price and quality) for each experiment.

Each of the above experiment is repeated 20 times ($T_{dead} = 20$) as the maximum number of negotiation rounds. For each negotiation round, the conducted owl search algorithm is repeated 25 times to generate the best offer. Table 4 lists the parameters of the COSA-based bilateral negotiation model in addition to the setting parameters for the compared algorithm PSO.

Table 3
Acceptable range of negotiation issues.

Product No.	The acceptable range of negotiation issue$i$	
	$[Min_p, Max_p]$	$[Min_q, Max_q]$
1	$[100, 400]$	$[30, 80]$
2	$[300, 750]$	$[50, 85]$
3	$[200, 700]$	$[35, 95]$
4	$[150, 480]$	$[40, 85]$
5	$[450, 800]$	$[50, 90]$
6	$[250, 500]$	$[45, 95]$
7	$[350, 750]$	$[50, 90]$
8	$[600, 900]$	$[55, 95]$
9	$[300, 600]$	$[45, 95]$
10	$[150, 350]$	$[35, 80]$

Table 4
OSA-based negotiation parameters and PSO.

Parameter	**value**
Number of negotiation rounds T_{dead}	20
Population size $\|P\|$	10
Number of iterations T_{max}	25
$\alpha,\ \beta$	$C(t)$
Acc_{low} (buyer/seller)	0.3
Acc_{high} (buyer/seller)	1
C_1 and C_2 (learning factors of PSO)	2

5.2. Performance metrics

The solution of the proposed algorithm against others is evaluated on the base of the following performance measures:

- Agreement Ratio (AR): This measure indicates the percentage of agreement negotiation cases concerning the total number of negotiations. It can be computed as follows

$$AR\ (\%) = \frac{Ag_b + Ag_s}{Neg_{total}}$$

Where Ag_b and Ag_s are the numbers of agreement by the buyer and seller side, respectively. Neg_{total} is the total number of negotiations. The higher the AR, the more agreement is reached, and both sides (buyer and seller) gain a better utility.

- Average buyer utility ($AvgU_b$): This measure indicates the average utility that buyers gain from the negotiation process; it can be computed according to the following.

$$AvgU_b = \sum_{i=1}^{Agr_{total}} UB_i / Agr_{total}$$

$$UB_i = RB_{b_i} - Acc_s$$

Where Agr_{total} is the total number of negotiation agreement; $Agr_{total} = Ag_b + Ag_s$, RB_{b_i} is the reserved price of buyer i and Acc_s is the accepted seller's offer price.

- Average seller utility ($AvgU_s$): This measure indicates the average utility that sellers gain from the negotiation process; it can be computed according to the following.

$$AvgU_s = \sum_{i=1}^{Agr_{total}} US_i / Agr_{total}$$

$$US_i = Acc_b - IB_{s_i}$$

Where Agr_{total} is the total number of negotiation agreement; $Agr_{total} = Ag_b + Ag_s$, IB_{s_i} is the open price of seller i and Acc_b is the accepted buyer's offer price.

- Average negotiation rounds ($AvgN$): This measure indicates the average number of negotiation rounds.

$$AvgN = \sum_{i=1}^{Agr_{total}} N_i / Agr_{total}$$

Where N_i is the total number of rounds required to reach a mutual agreement for negotiation process i between a buyer and a seller.

- Average processing time ($AvgT$): This measure indicates the average time (in seconds) required to reach an acceptance between negotiation parties.

Table 5
Experimental results at (buyer side b).

Exp. #	**PSO**				**OSA**				**COSA**			
	Min.	Max.	Avg.	Std.	Min.	Max.	Avg.	Std.	Min.	Max.	Avg.	Std.
1	0.236	0.296	0.271	0.017	0.213	0.269	0.247	0.018	0.177	0.225	**0.198**	0.013
2	0.276	0.31	0.298	0.011	0.246	0.291	0.274	0.014	0.208	0.253	**0.231**	0.014
3	0.239	0.274	0.258	0.011	0.215	0.249	0.233	0.012	0.181	0.219	**0.2**	0.01
4	0.286	0.317	0.301	0.01	0.258	0.291	0.276	0.011	0.235	0.263	**0.246**	0.008
5	0.307	0.35	0.33	0.013	0.227	0.322	0.3	0.021	0.221	0.276	**0.251**	0.015

Please cite this article as: W. H. El-Ashmawi, D. S. Abd Elminaam, A. M. Nabil et al., A chaotic owl search algorithm based bilateral negotiation model, Ain Shams Engineering Journal, https://doi.org/10.1016/j.asej.2020.01.005

Table 6
Experimental results at (seller side s).

Exp. #	PSO				OSA				COSA			
	Min.	Max.	Avg.	Std.	Min.	Max.	Avg.	Std.	Min.	Max.	Avg.	Std.
1	0.244	0.37	0.276	0.028	0.209	0.358	0.253	0.031	0.176	0.316	**0.205**	0.031
2	0.278	0.42	0.309	0.036	0.252	0.413	0.287	0.039	0.21	0.413	**0.245**	0.049
3	0.244	0.391	0.27	0.037	0.218	0.379	0.246	0.041	0.181	0.373	**0.213**	0.046
4	0.286	0.34	0.306	0.014	0.26	0.327	0.281	0.016	0.235	0.321	**0.252**	0.021
5	0.308	0.362	0.331	0.014	0.225	0.353	0.303	0.025	0.225	0.323	**0.255**	0.021

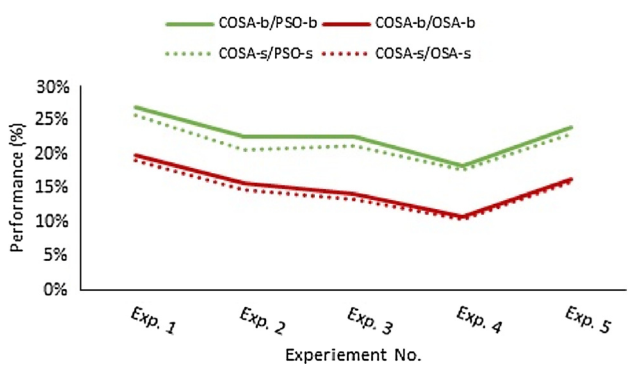


Fig. 6. Performance (%) of COSA at Buyer/Seller sides.

$$AvgT = \sum_{i=1}^{Ag_b+Ag_s} T_i / Agr_{total}$$

Where T_i is the time of the negation process i to reach acceptance.

- Average buyer fitness value ($AvgF_b$): This measure indicates the average fitness value at the buyer side for generating a feasible offer during iterations of the meta-heuristic algorithm. It can be computed as follow

$$AvgF_b = \sum_{i=1}^{N} f_{b_i}(O)/N$$

Where $f_{b_i}(O)$ is the buyer's global best fitness value of the i^{th} negotiation round.

- Average seller fitness value ($AvgF_s$): this measure indicates the average fitness value at the seller side for generating a feasible offer during iterations of meta-heuristic algorithms. It can be computed as follow.

$$AvgF_s = \sum_{i=1}^{N} f_{s_i}(O)/N$$

Where $f_{s_i}(O)$ is the seller's global best fitness value of the i^{th} negotiation round.

Moreover, the improvement of fitness value in terms of minimizing the difference between the offer and counter-offer at both sides of the negotiation of the proposed algorithm against other algorithms can be computed as follows.

$$Improvement\ (\%) = \frac{(f_{alg} - f_{COSA})}{f_{alg}} * 100$$

Where $alg = \{OSA, PSO\}$, f_{alg} is the fitness value of each of the existing algorithms and f_{COSA} is the fitness value result of the proposed algorithm.

For computing the performance of the proposed algorithm during iterations, the following formula can be used.

$$Improvement\ performance\ (\%) = \frac{(f_i - f_j)}{f_i} * 100$$

Where f_j is the fitness value of the proposed solution at j^{th} iteration number and j is any subsequent iteration number (i.e., $j > i$).

5.3. Experimental results

Tables 5 and 6 show the experimental results of meta-heuristic algorithms (PSO, OSA, and COSA) at both negotiation sides (buyer b and seller s), respectively. The maximum (Max.), minimum (Min.), average (Avg.) and standard deviation (Std.). The best results in the table are **boldfaced**. The efficiency of the proposed COSA appears in the average results of fitness value over negotiation rounds.

For example, as in Exp.2, the $AvgF_b$ of the proposed algorithm COSA is **0.231** while $AvgF_b$ of OSA and PSO are 0.274 and 0.298, respectively. It was clear from the obtained results at both sides of negotiation that COSA has achieved a minimum average fitness value. The average results values are taken to ensure the accuracy of the proposed model.

Fig. 6 illustrates the achieved performance of the proposed COSA over PSO and standard OSA in both sides of negotiation (i.e., buyer (b) and seller (s)) in terms of Avg. fitness value.

At the buyer side, the proposed COSA-b has achieved a better performance in terms of average fitness values than PSO-b within a range from 18% to 27% throughout different experiments. When compared with standard CSA-b, the chaotic OSA at the buyer side has gained more performance than CSA-b ranged from 11% to 20% for different test experiments. At the seller side, the performance of the COSA-s has reached up to 26% over PSO-s and up to 19% over standard OSA-s in terms of average fitness value.

Table 7
Performance of COSA.

Exp. #	**Performance (in terms of Min. fitness value)**			
	Buyer side(b)		Seller side(S)	
	COSA/PSO	COSA/OSA	COSA/PSO	COSA/OSA
Exp. 1	25%	17%	28%	16%
Exp. 2	25%	15%	24%	17%
Exp. 3	24%	16%	26%	17%
Exp. 4	18%	9%	18%	10%
Exp. 5	28%	3%	27%	0%

Please cite this article as: W. H. El-Ashmawi, D. S. Abd Elminaam, A. M. Nabil et al., A chaotic owl search algorithm based bilateral negotiation model, Ain Shams Engineering Journal, https://doi.org/10.1016/j.asej.2020.01.005

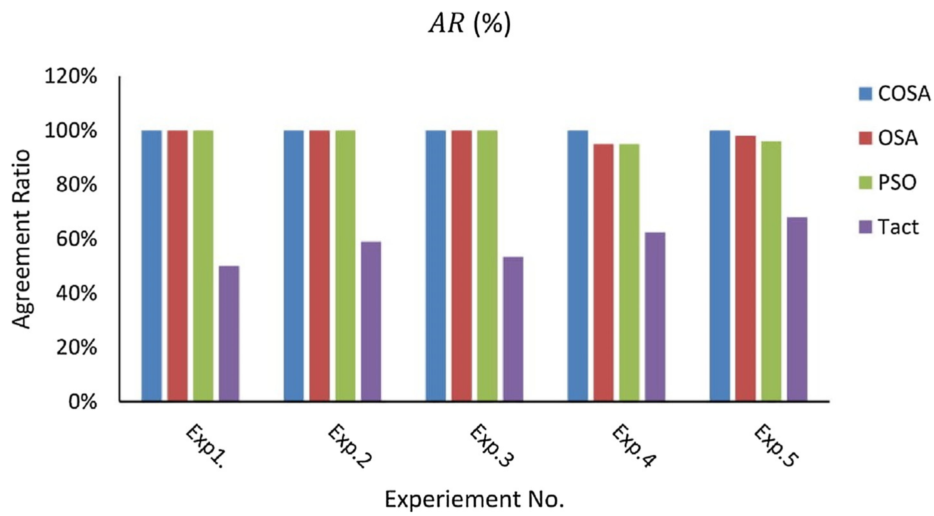


Fig. 7. Agreement ratio.

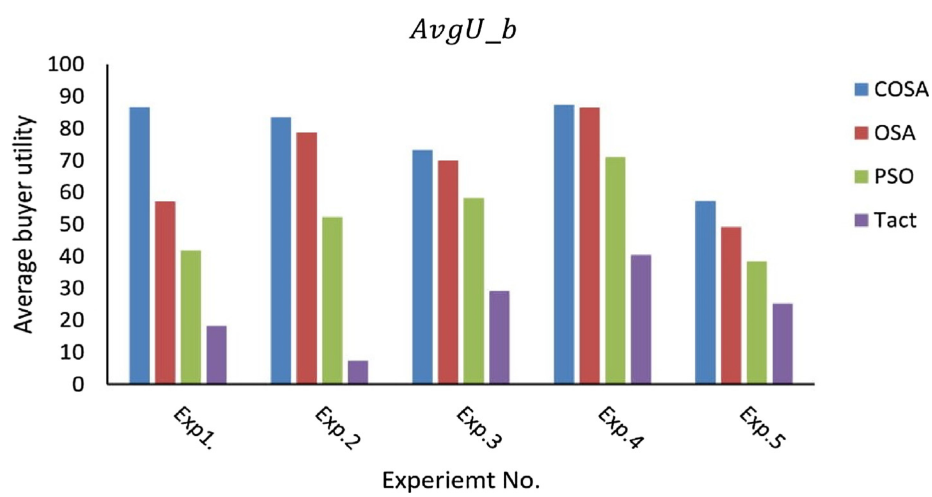


Fig. 8. Average buyers' utility.

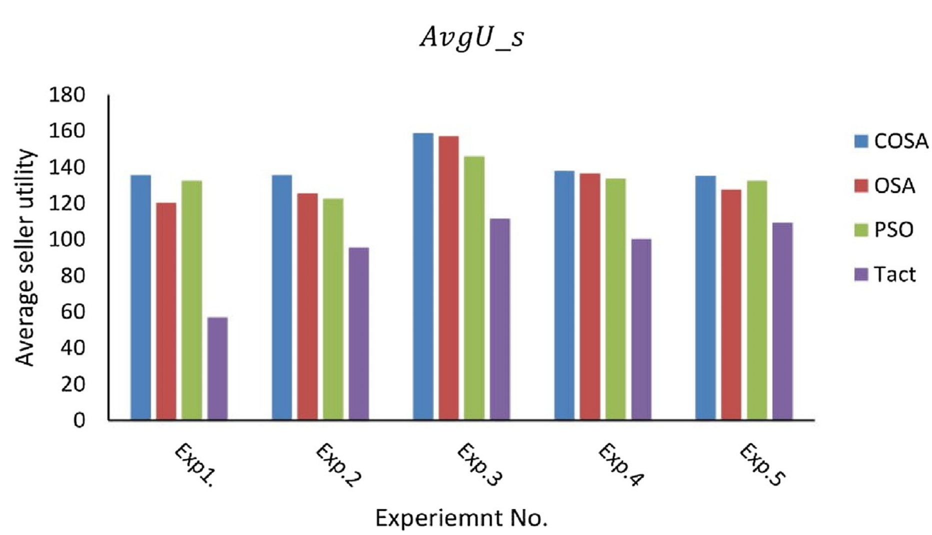


Fig. 9. Average sellers' utility.

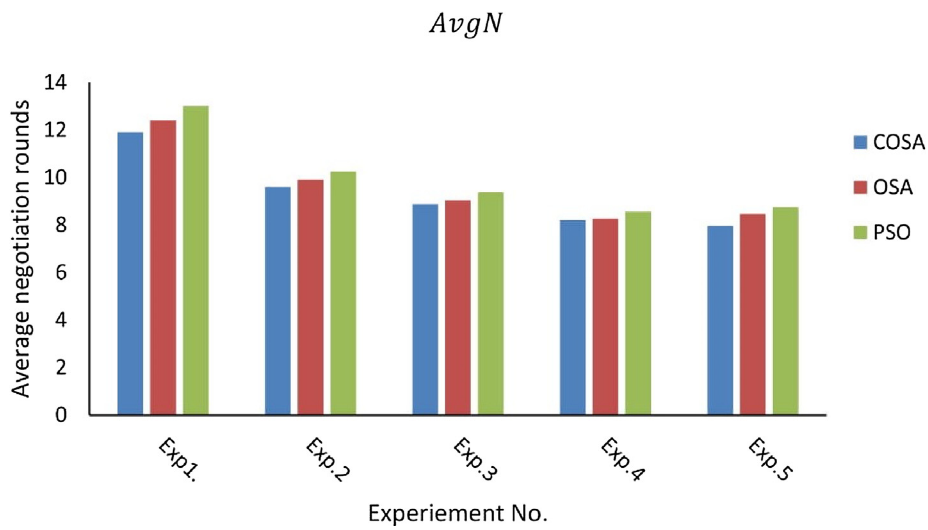


Fig. 10. Average negotiation rounds.

As well, the performance of the proposed algorithm in terms of the Min. fitness value achieved is better than PSO and OSA as listed in Table 7 for different experiements.

From the above table, the COSA has achieved better performance in terms of minimum fitness value reached up to 28% at both buyer and seller sides when compared with PSO. Regarding the standard OSA, the integration of chaos with OSA has gained better performance reached up to 17% at both sides of the negotiation.

Additionally, the performance analysis of COSA against others in terms of agreement ratio is shown in Fig. 7. The figure illustrated the number of agreement cases out of whole negotiation cases for all algorithms throughout various experiments to obtain a successful agreement between negotiation parties.

From Fig. 7, the proposed algorithm reached a successful agreement in all experiments, while the OSA and PSO have improved up to 95% in some experiments. On the contrary, the tactical decision function agreement ranged from 50% to 68%.

Figs. 8 and 9 illustrated the average buyer and seller utility, respectively that gain from the negotiation process.

From the figure, the proposed algorithm has obtained a higher average utility for buyers in all experiments when compared with other comparative algorithms. The improvement of COSA results is better than OSA by 34%, better than PSO by 52%, and better than Tact up to 79% throughout different experiments.

As a result of the negotiation process, the proposed algorithm has achieved the best average seller utility when compared with other algorithms as shown in Fig. 9. It has obtained average utility ranged from 6% to 11% better than OSA, ranged from 2% to 10% better than PSO and ranged from 19% to 58% better than tact. However, the PSO has obtained average seller utility better than OSA with 9% and 4% for experiments 1 and 5, respectively.

The proposed algorithm COSA based negotiation has proven its effectiveness (i.e., regarding the average buyer and seller utility) and its efficiency (i.e., regarding the negotiation rounds and processing time), as seen in Figs. 10 and 11.

Fig. 10 illustrated the average number of negotiation rounds for acceptance (in agreement case only). Although the chaotic OSA has lower results than others, it has nearly resulted in OSA in some experiments (e.g., Exp. 3 and Exp. 4). The average number of rounds of the proposed algorithm reached up to 9% lower than PSO and 6% lower than OSA.

From Fig. 11, the average processing time for negotiation agreement ranged from 15 to 23 s in the case of COSA, while in case of

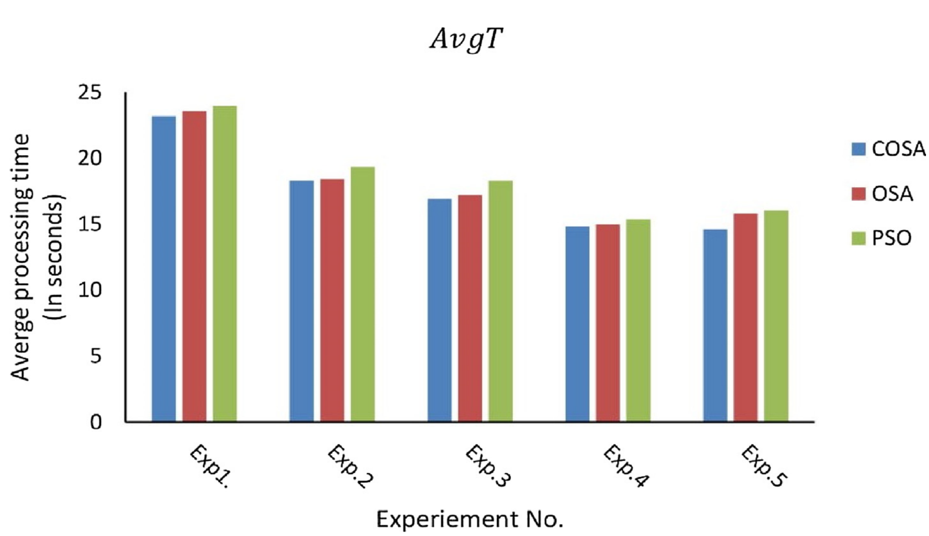


Fig. 11. Average processing time.

Please cite this article as: W. H. El-Ashmawi, D. S. Abd Elminaam, A. M. Nabil et al., A chaotic owl search algorithm based bilateral negotiation model, Ain Shams Engineering Journal, https://doi.org/10.1016/j.asej.2020.01.005

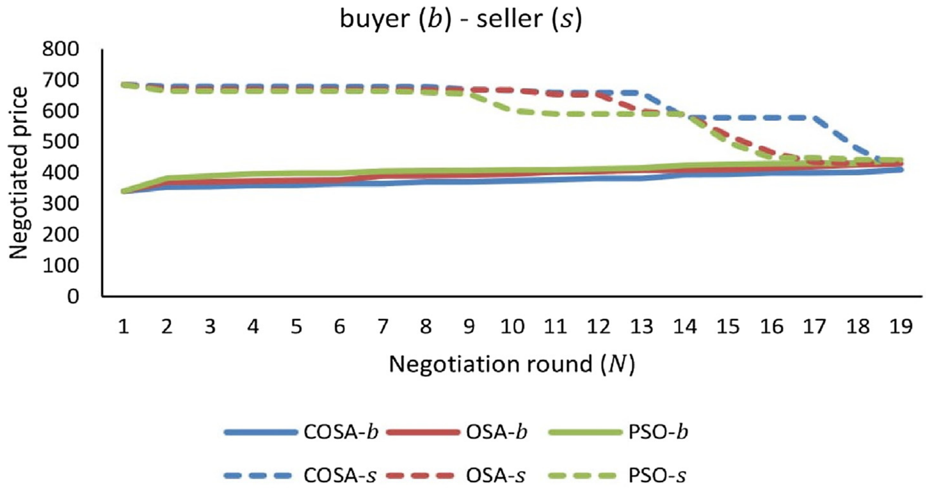


Fig. 12. Buyer-Seller negotiated the price.

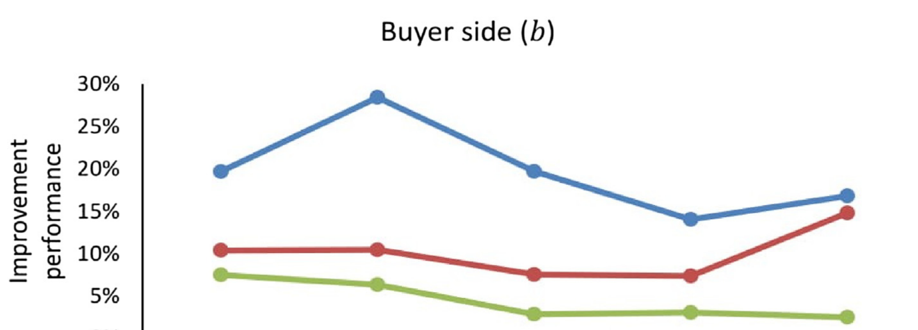
OSA and PSO, it ranged from 15 to 24 s. Throughout experiments, the proposed algorithm has achieved average processing time minimized by 8% when compared with OSA and 9% when compared with PSO.

Fig. 12 shows the concession behavior of the buyer/seller offer throughout the sequence of negotiation rounds. This figure illustrated an example of buyer-seller negotiation price in order to reach a mutual agreement before the deadline reached.

In all matching algorithms from Fig. 12, the buyer starts with his minimum price (e.g., 340) and the seller starts with his maximum price (e.g., 685). During negotiation process, the buyer increases his price, and the seller decreases his price until a mutual agreement has been reached (e.g., agreement zone). From the above figure, the chaotic owl search algorithm based negotiation at buyer side (COSA-*b*) increases the price adequately concerning the number of negotiation rounds. While in OSA-*b* and PSO-*b*, the price increases by a more significant amount than COSA-*b*. OSA-*b* price increased within range 3%~6% more than COSA while PSO-*b* price increase within range 7%~10% more than COSA during negotiation rounds. On the other side (i.e., seller), OSA-*s* and PSO-*s* decrease the price dramatically concerning negotiation deadline in order to achieve a mutual agreement. The OSA-*s* price decreases up to 25%, and PSO-*s* price decreases up to 22% during iteration when compared with COSA-*s*. Moreover, the proposed algorithm tries to gain a maximum utility not only for a buyer side but also for a seller side by decreasing the negotiated price effectually. Typically, the two negotiated parties can continue negotiation and exchange offers until an agreement is reached or a pre-specified negotiation deadline (e.g., number of rounds) is passed. In this case, both buyer and seller reached agreement on accepted price **409** in case of the proposed algorithm, while in case of OSA, the accepted price is 429 and 441 for PSO. The concession process of COSA-*s* is very adaptive and sensitive to deadline as it decreases rapidly as the deadline reached.

The maximal improvement performance of the proposed COSA and other compared algorithms for both buyer and seller is shown in Figs. 13 and 14, respectively.

From the above figures and tested results, the proposed algorithm has achieved an average buyer/seller utility better than other compared algorithms due to the great combination between chaotic theory and the owl search algorithm which leads the algorithm to explore the search space effectively and locate an optimal solution for both negotiation parties.



	Exp1.	Exp.2	Exp.3	Exp.4	Exp.5
COSA	20%	28%	20%	14%	17%
OSA	10%	10%	8%	7%	15%
PSO	8%	6%	3%	3%	3%

Experiement No.

Fig. 13. Improvement performance at the buyer side.

Fig. 14. Improvement performance at the seller side.

6. Conclusion

In most real-world negotiation environment, negotiation parties with different goals jointly search for a mutual solution that maximizes their utilities. Due to extensive negotiation space in terms of a number of offers, multi-issue negotiation and limited information about negotiators, meta-heuristic optimization algorithms can play a vital role to evaluate these offers and provide a feasible one. This paper focuses on a multi-issue bilateral negation model based on one of the recently meta-heuristic algorithms (Owl search algorithm) for evaluating offers towards generating new offers (counter-offer). The proposed model built upon the integration of logistic chaotic logistic map into an owl search algorithm to enhance the searching capability for agreement. This integration presents a novelty step towards bilateral negotiation that results in mutual benefits for both negotiation parties. The proposed chaotic owl search algorithm (COSA) has two main concerns, the first one is the feasible initial population of offers closer to the goal offer. The second concerns based on the updating movement of owls through the integration of the logistic chaotic map. This integration has a great influence on the changed intensity of current owl and the opposite offer, respectively. various experiments consist of several buyers, several sellers over a set of negotiated issues are conducted to demonstrate the effectiveness of the proposed COSA in improving the quality of the negotiation process. Additionally, it compared with other existing algorithms such as PSO, standard OSA and tactics. The proposed COSA has achieved average fitness value better than PSO within a range from 18% to 27% at the buyer side and up to 26% at the seller side. Moreover, it beats the standard OSA at both sides up to 17% in terms of minimum fitness value throughout different experiments. For agreement ratio, the COSA obtains a successful agreement in all negotiation cases while PSO and OSA have obtained 95% and tactical has obtained agreement ratio ranged from 50% to 68%. In addition, it has the superiority in terms of average buyer and seller utility with 34% (i.e., at buyer side) better than standard OSA and 11% (i.e., at seller side) and 52% (i.e., at buyer side) better than PSO and 10% (i.e., at seller side). The mutual agreement between negotiators has been achieved through a lower number of negotiation rounds by 9% than PSO and 6% than OSA. For average processing time, the proposed COSA beats other compared algorithms due to the integration of chaos for better exploration of the search space. From these concluded results, the COSA gains The maximal improvement performance for improving the quality of the negotiation process. Some future recommendations are proposed, which involve modelling a multi-lateral negotiation. The multi-lateral negotiation can be modelled as a multi bilateral negotiation which consists of more than one negotiation process at a time and large scale number of negotiation parties. Another direction is the integration of different chaotic maps that may have a big influence on the search space for agreements.

References

[1] Wang Gong, Wong TN, Yu Chunxia. A computational model for multi-agent E-commerce negotiations with adaptive negotiation behaviors. J Comput Sci 2013;4(3):135–43. doi: https://doi.org/10.1016/j.jocs.2011.10.003.
[2] Gerding EH, Bragt DDB, La Poutré JA. Scientific approaches and techniques for negotiation: A game-theoretic and artificial intelligence perspective. Amsterdam, The Netherlands: Technical report CWI (Centre for Mathematics and Computer Science); 2000.
[3] Jennings NR, Faratin P, Lomuscio AR, Parsons S, Wooldridge MJ, Sierra C. Automated negotiation: Prospects, methods, and challenges. Group Decis Negot 2001;10(2):199–215.
[4] Kraus S. Strategic negotiation in multiagent environments. Cambridge, MA: MIT Press; 2001.
[5] Li C, Giampapa J, Sycara KP. A review of research literature on bilateral negotiations. Robotics Institute, Pittsburgh, PA: Technical report; 2003.
[6] Silaghi GC, Serban LD, Litan CM. A framework for building intelligent SLA negotiation strategies under time constraints. In: Altmann J, Rana OF, editors. Proceedings of economics of grids, clouds, systems, and services: 7th international workshop, Vol. 6296. New York: SpringerVerlag Inc; 2010. p. 48.
[7] Binmore K, Vulkan N. Applying game theory to automated negotiation. Netnomics 1999;1(1):1–9.
[8] Liang Y-Q, Yuan Y. Co-evolutionary stability in the alternating-offer negotiation. In: IEEE conference on cybernetics and intelligent systems. p. 1176–80.
[9] Fatima S, Wooldridge M, Jennings NR. Comparing equilibria for game-theoretic and evolutionary bargaining models. In: Proceedings of the International Workshop on Agent-Mediated Electronic Commerce V, Melbourne, Australia. p. 70–7.
[10] He M, Jennings NR, Leung H. On agent-mediated electronic commerce. IEEE Trans Knowl Data Eng 2003;15(4):985–1003.
[11] Osborne MJ, Rubinstein A. Bargaining and Markets (Economic theory, econometrics, and mathematical economics). New York: Academic Press; 1990.
[12] Lang Fabian, Fink Andreas. Learning from the metaheuristics: protocols for automated negotiations. Group Decis Negot 2014;24:299–332. doi: https://doi.org/10.1007/s10726-014-9390-x.
[13] Wang Gong, Wong TN, Chunxia Yu. Computational method for agent-based E-commerce negotiations with adaptive negotiation behaviors. Procedia Comp Sci 2011;4:1834–43.
[14] Louta M, Roussaki I, Pechlivanos L. An intelligent agent negotiation strategy in the electronic marketplace environment. Eur J Oper Res 2008;187(3):1327–45.
[15] Chen YM, Huang PN. Agent-based bilateral multi-issue negotiation scheme for e-market transactions. Appl Soft Comput 2009;9(3):1057–67.
[16] Lau RYK. Towards a web services and intelligent agents-based negotiation system for B2B eCommerce. Electron Commer Res Appl 2007;6(3):260–73.
[17] Lee CC, Ferguson MJ. To reveal or not to reveal? Strategic disclosure of private information in negotiation. Eur J Oper Res 2010;207(1):380–90.

Please cite this article as: W. H. El-Ashmawi, D. S. Abd Elminaam, A. M. Nabil et al., A chaotic owl search algorithm based bilateral negotiation model, Ain Shams Engineering Journal, https://doi.org/10.1016/j.asej.2020.01.005

[18] Zhang LL, Song HG, Chen XG, Liu H. A simultaneous multi-issue negotiation through autonomous agents. Eur J Oper Res 2011;210(1):95–105.
[19] Zhang LL, Chen XG. An agent-based multi-attribute sealed-bid design for bilateral contract. J Softw 2009;4(1):65–72.
[20] Lai GM, Li CH, Sycara K. Efficient multi-attribute negotiation with incomplete information. Group Decis Negot 2006;15(5):511–28.
[21] Jennings NR, Faratin P, Lomuscio AR, et al. Automated negotiation: Prospects, methods, and challenges. Group Decis Negot 2001;10(2):199–215.
[22] Da-Jun C, Liang-Xian X. A negotiation model of incomplete information under time constraints. In: Proceedings of the International Conference on Autonomous Agents and Multiagent Systems. p. 128–34.
[23] Fatima SS, Wooldridge M, Jennings N. Bargaining with incomplete information. Ann Math Artif Intell 2005;44(3):207–32.
[24] Narayanan Vidya, Jennings NR. An adaptive bilateral negotiation model for e-commerce settings. In: Seventh IEEE International Conference on E-Commerce Technology (CEC'05), Munich, Germany. p. 34–41. doi: https://doi.org/10.1109/ICECT.2005.15.
[25] Robu V, Somefun DJA, La Poutré JA. Modeling complex multi-issue negotiations using utility graphs. In: Proceedings of the International Conference on Autonomous Agents and Multiagent Systems, New York, USA. p. 280–7.
[26] Von Neumann J, Morgenstern O. The theory of games and economic behaviour. Princeton: Princeton University Press; 1994.
[27] Barbuceanu M, Lo W. A multi-attribute utility theoretic negotiation architecture for electronic commerce. In: Proceedings of 4th Int Conf. on Autonomous Agents Barcelona, Spain. p. 239–47.
[28] Krovi R, Graesser A, Pracht W. Agent behaviors in virtual negotiation environments. IEEE Trans Syst, Man, Cybernet 1999;29(1):15–25.
[29] Matwin S, Szapiro T, Haigh K. Genetic algorithm approach to a negotiation support system. IEEE Trans Syst Man Cybernet 1991;21(1):102–14.
[30] Rubenstein-Montano B, Malaga RA. A weighted sum genetic algorithm to support multiple-party multi-objective negotiations. IEEE Trans Evol Comput 2002;6(4):366–77.
[31] Lau RYK. Towards genetically optimized multi-agent multi-issue negotiations. In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS '05). p. ..
[32] Jonker C, van der Meij L, Robu V, Treur J. Demonstration of a software system for automated multi-attribute negotiation. Proceedings of the International Conference on Autonomous Agents and Multiagent Systems, New York City, USA, 2004.
[33] Baarslag Tim, Hendrikx Mark JC, Hindriks Koen V, Jonker Catholijn M. Learning about the opponent in automated bilateral negotiation: a comprehensive survey of opponent modeling techniques. Autonom Agents Multi-Agent Syst 2016;30(5, 30):849–98. doi: https://doi.org/10.1007/s10458-015-9309-1.
[34] Zhang LL, Chen XG, Liu H. Optimality strategy of a sealed-offer simultaneous bargaining protocol. Sci China Ser F: Inform Sci 2011;54(1):79–90.
[35] Linlan ZHANG, Qing LIU, Huaming GUI. An automated bilateral negotiation about price and quantity in E-commerce. Appl Mech Mater 2013;411–414:2219–22. doi: https://doi.org/10.4028/www.scientific.net/AMM.411-414.2219.
[36] Zhang Linlan, Liu Qing. An automated multi-issue negotiation mechanism based on intelligent agents in E-commerce. JOAMS 2016:172–5. doi: https://doi.org/10.12720/joams10.12720/joams.4.2.172-175.
[37] Blum Christian, Roli Andrea. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Comput Surv 2003;35(3):268–308. doi: https://doi.org/10.1145/93750310.1145/937503.937505.
[38] Klein M, Faratin P, Sayama H, Bar-Yam Y. Negotiating complex contracts. Group Decis Negot 2003;12(2):111–25.
[39] Klein M, Faratin P, Sayama H, Bar-Yam Y. Negotiating complex contracts. MIT Sloan School of Management Working Paper No. 4196-01. Cambridge: Massachusetts Institute of Technology; 2007.
[40] Fink A. Supply chain coordination by means of automated negotiations between autonomous agents. In: Chaib-draa B, Müller J, editors. Multiagent based supply chain management (Studies in Computational Intelligence, Vol. 28. Berlin: Springer; 2006. p. 351–72.
[41] Kui F, Guihua N. Application of particle swarm optimization algorithm in E-commerce negotiation. The 6th Wuhan International Conference on E-Business. p. 225–30.
[42] Ali R, El Bakrawy LM, Ghali NI. An optimized approach for E-commerce negotiation. Int J Electron Commun Comp Eng (IJECCE) 2013;4(1):142–5.
[43] Silva F, Faia R, Pinto T, Praça I, Vale Z. Optimizing opponents selection in bilateral contracts negotiation with particle swarm. In: Bajo J, editor. Highlights of Practical Applications of Agents, Multi-Agent Systems, and Complexity: The PAAMS Collection. PAAMS 2018. Communications in Computer and Information Science, vol 887. Cham: Springer; 2018.
[44] Esmaeili, Ahmad & Mozayani, Nasser. Improving multi-agent negotiations using multi-objective PSO algorithm. 2010;92–101. 10.1007/978-3-642-13480-7_11.
[45] Rui Tang, Simon Fong, and Nilanjan Dey (March 28th 2018). Metaheuristics and Chaos Theory, Chaos Theory, Kais A. Mohamedamen Al Naimee, IntechOpen, http://doi.og/10.5772/intechopen.72103. Available from: https://www.intechopen.com/books/chaos-theory/metaheuristics-and-chaos-theory.
[46] Assarzadeh Z, Naghsh-Nilchi AR. Chaotic particle swarm optimization with mutation for classification. J Med Signals Sens 2015;5(1):12.
[47] Sheikholeslami R, Kaveh A. A survey of chaos embedded metaheuristic algorithms. Int J Optim Civil Eng 2013;3(4):617–33.
[48] Alatas B, Akin E, Ozer AB. Chaos embedded particle swarm optimization algorithms. Chaos Soliton Fract 2009;40:1715–34.
[49] Alatas B. Chaotic harmony search algorithm. Appl Math Comput 2010;29(4):2687–99.
[50] Rosenschein JS, Zlotkin G. Rules of encounter: designing conventions for automated negotiation among computers. Cambridge: MIT Press; 1994.
[51] Faratin P, Sierra C, Jennings NR. Negotiation decision functions for autonomous agents. Int J Robot Autonom Syst 1998;24(3–4):159–82.
[52] Gupta Aakanksha, Srivastava Durgesh Kumar, Jain Saket. Evaluating negotiation protocols and negotiation strategies for automated E-commerce. Int J Eng Res Technol (IJERT) 2016;5(07):502–5.
[53] Krause DR, Terpend R, Petersen KJ. Bargaining stances and outcomes in buyer-seller negotiations: experimental results. J Supply Chain Manag 2006;42(3):4–15.
[54] Rubinstein A. Perfect equilibrium in a bargaining model. Econometrica 1982;50:97–109.
[55] Ragone A, Di Noia T, Di Sciascio E, Donini FM. Alternating-offers protocol for multi-issue bilateral negotiation in semantic-enabled marketplaces. In: Aberer K, editor. The Semantic Web. ISWC, 2007; ASWC 2007. Lecture Notes in Computer Science, vol. 4825. Berlin, Heidelberg: Springer; 2007.
[56] Faratin P. Automated service negotiation between autonomous computational agents Ph.D. Thesis. University of London; 2000.
[57] Jain M, Maurya S, Rani A, Singh V. Owl search algorithm: a novel nature-inspired heuristic paradigm for global optimization. J Intell Fuzzy Syst 2018;34:1573–82.
[58] https://en.wikipedia.org/wiki/Owl, Retrieved February 2018.
[59] Grothe B. How the barn Owl computes auditory space. Trends Neurosci 2018;41(3):115–7. doi: https://doi.org/10.1016/j.tins.2018.01.004.
[60] Gutiérrez-Ibáñez Cristián, Iwaniuk Andrew, Wylie Douglas. Relative size of auditory pathways in symmetrically and asymmetrically eared Owls. Brain Behav Evol 2011;78:286–301. doi: https://doi.org/10.1159/000330359.
[61] Carr Catherine E, Christensen-Dalsgaard Jakob. Sound localization strategies in three predators. Brain, Behav Evol 2015;86(1):17–27.
[62] http://hyperphysics.phy-astr.gsu.edu/hbase/Acoustic/invsqs.html. Retrieved February 2018.
[63] Faratin P, Sierra C, Jennings NR. Using similarity criteria to make issue trade-offs in automated negotiations. Artif Intell 2002;142(2):205–37.
[64] Kennedy J. Particle Swarm Optimization. US: Encyclopedia of Machine Learning. Springer; 2010. p. 760–6.

Walaa H. El-Ashmawi is an Assistant Professor at Suez Canal University, Faculty of Computers and Informatics, Computer Science Department. Egypt. She received her B.Sc. in 2001 and M.Sc. in 2008 from Egypt. She got her PhD. Degree in 2013 from the Computer Science Department, College of Information Science and Engineering, Hunan University, China. She authors and coauthors to many journal publications and conferences in the area of Artificial Intelligence, Team formation problem, Multi-agent Systems, Expert systems and Neural Networks.

Diaa Salama Abd Elminaam was born on November 23, 1982, in KafrSakr, Sharkia, Egypt. He received the B.Sc from Faculty of Computers &Informatics, Zagazig University, Egypt in 2004 with grade very good with honour, and obtains the master degree in information system from the faculty of computers and information, menufia university, Egypt in 2009 specialising in Cryptography and network security. He obtained his PhD degree in information system from the faculty of computers and information, menufia university, Egypt in 2015. He is currently an Assistance Professor in Information systems department, Faculty of Computers and Information, Benha University, Egypt since 2011. He has worked on several research topics. .Diaa has contributed more than 50+ technical papers in the areas of wireless networks, wireless network security, Information security and Internet applications, Cloud Computing, Mobile Cloud Computing, Internet of Things, and Machine learning in international journals, international conferences, local journals and local conferences. He majors in Cryptography, Network Security, IoT, Big Data, Cloud Computing, deep learning.)

Ayman M. Nabil, is an Egyptian Researcher who lives in Cairo, Egypt. He got his Phd in Computer Science from Helwan university on 2011. He is working as an Assistant Professor, Acting Head of Computer Science Department in Misr international University. His research interests are in many Fields, like: image Processing – Pattern Recognition – Machine Learning – Remote Sensing -Bioinformatics

Esraa Eldesouky, She is an assistant professor in College of Computer Engineering and Sciences at Prince Sattam Bin Abdulaziz University. She finished her Ph.D. in College of Information Science and Engineering at Hunan University in 2015. She earned the M.Sc. degree in computer science and technology from Hunan University (2011). The current research interest includes VANETs, content handling, intelligent transportation systems, and coalition formation game

Please cite this article as: W. H. El-Ashmawi, D. S. Abd Elminaam, A. M. Nabil et al., A chaotic owl search algorithm based bilateral negotiation model, Ain Shams Engineering Journal, https://doi.org/10.1016/j.asej.2020.01.005