Available online at www.sciencedirect.com

SciVerse ScienceDirect

Physics Procedia 25 (2012) 1666 – 1670

Physics Procedia

2012 International Conference on Solid State Devices and Materials Science

A Winner Determination Algorithm for Combinatorial Auctions Based on Hybrid Artificial Fish Swarm Algorithm

Genrang Zheng[a],ZhengChun Lin[b]

[a]*Department of Computer Engineering,Zhongshan Polytechnic,Zhongshan, China*
[b]*Department of Research & Development,Foshan Institute of Standards Technology,Foshan, China*

Abstract

The problem of winner determination in combinatorial auctions is a hotspot electronic business, and a NP hard problem. A Hybrid Artificial Fish Swarm Algorithm(HAFSA), which is combined with First Suite Heuristic Algorithm (FSHA) and Artificial Fish Swarm Algorithm (AFSA), is proposed to solve the problem after probing it base on the theories of AFSA. Experiment results show that the HAFSA is a rapidly and efficient algorithm for The problem of winner determining. Compared with Ant colony Optimization Algorithm, it has a good performance with broad and prosperous application.

Keywords-combinatorial auctions; winner determination; Artificial Fish Swarm Algorithm

1.Introduction

Combinatorial auctions have a wide range of applications, and they are prevalent in the field of artificial intelligence. Many problems can be abstract into a combinatorial auction problem. Combinatorial auctions allow bidders to place bids on bundles of items, and leads to more efficient allocations than traditional auction mechanism. Determining the winners in combinatorial auctions is NP-Complete problem, so the exact solutions of combinatorial auctions are hard to obtaind. The problem can be described as follows[1]: Let M is the set of indivisible items to sell, $M = (1, 2, \cdots, m)$, B is the set of bidders, $B = (b_1, b_2, \cdots, b_n)$. A bidder b_i can bid a single item of M or a combination of a number of items S, where S is a nonempty subset of items M. $b_i(S)$ is the price that b_i will offer for the items set S. The winner determination problem is to find a set of bids to maximize the auctioneer's revenue under the condition that each item only can be allocated one time [2-5]. A seller only want the maximum price bid during bidders preprocessing and discards other bids. The maximum price bid can be wrote as

 doi:10.1016/j.phpro.2012.03.292

$$b(S) = \underset{i \in bidders}{MAX}\, b_i(S). \tag{1}$$

The bids described below are preprocessed, then the model of winner determination problems in combinatorial auctions can be described as follows [6],

$$\underset{X \in A}{MAX} \sum_{S \in X} b(S). \tag{2}$$

The object of the model is to maximize the auctioneer's revenue. Each X is a feasible solution, each item only can be allocated one time. A is a set of feasible solutions, $A = \{W \in \Gamma | S, S' \in W \Rightarrow S \cap S' = \Phi\}$, and $\bigcup_{S \in W} S \subseteq M$, Γ is a subset of W.

As mentioned above, the winner determination problem is a discrete combinatorial optimization problem, and a NP-Complete problem, so the solving processing is very complex.

Artificial Fish-Swarm Algorithm (AFSA) [7-8] is a bionic optimization algorithm based on the research on the intelligent behavior of a swarm. AFSA has been concerned by researchers since it proposed. Many improved methods have been proposed and applied widely, for example, an improved AFSA based on girding method [9], an improved AFSA for solving numerical derivative [10], AFSA for signal MP-based sparse decomposition [11], AFSA applied in active shape models [12], in machining path planning of robot [13], and in range interval optimization of prestressed anchors [14], etc.

The fish can find nutrient places by themselves or following other fish in water. So the places where contains the largest number of fish always are the most nutrient. According to these facts, AFSF imitates the behaviors such as looking for food, clustering and trailing, and some random behaviors to find the optimal solution by constructing artificial fish swarms. Some commonly behaviors of fish are as follows:

- Behavior of looking for food: In general, the fish are randomly and freely moving in the water. As feeding the fish, they will gradually move to the place where food is increasing.
- Clustering behavior: In order to survive and avoid hazards, the fish will naturally clustered. There are three rules while fish clustering: Firstly, a fish will try to keep a certain distance with each other to avoid crowding; Secondly, a fish will try to move in a similar direction with its surrounding partners; Finally, a fish will try to move to the center of its surrounding partners.
- Trailing behavior: When some fish find food, other fish will follow these fish to find the food.
- Random behavior: Sometimes a fish swims randomly and freely to find food in water.

2.Improved AFSA for Combinatorial Auctions

Assuming that the dimension of the searching space is n corresponding to the number of bidders, the size of the school of fish is N. An artificial fish can be denoted as an N-dimensional vector (fish vector) $\mathbf{X}_i = (x_{i1}, x_{i2}, \cdots, x_{in})$, $(i = 1, 2, \cdots, N)$, the fitness function $Y = f(\mathbf{X})$ describes the current food concentration where a fish $\mathbf{X}$ in. $d_{i,j} = d(\mathbf{X}_i, \mathbf{X}_j)$ is the distance between artificial fish $\mathbf{X}_i$ and $\mathbf{X}_j$, δ is the factor of crowding degree. N_{try} denotes the maximum number of the artificial fish try moving. R_{vis} denotes the range of the artificial fish's vision.

2.1.Initialization

We randomly initialize the school of fish. As mentioned above, M is the set of indivisible items to sell, $M = (1, 2, \cdots, m)$. The N-dimensional vector or the fish vector $\mathbf{X}_i = (x_{i1}, x_{i2}, \cdots, x_{in})$ corresponding to an artificial fish is a random sequence (fish sequence), where $x_{ij} \in \{1, 2, \cdots, n\}$, $\forall j \neq j' \Rightarrow x_{ij} \neq x_{ij'}$.

2.2.Fitness Calculation

The value of the sum of all items will be obtained by the fitness function. An algorithm (Priority Fit Heuristic Algorithm, PFHA) is designed to obtain the fitness. PFHA are detailed as follows:

Step 1: Select the first bidder $b_i(S_i)$ as the first number of the fish sequence, let $U = \{S_i\}$.

Step 2: Select a bidder $b_j(S_j)$ from the rest bidders. If $U \cap S_j = \Phi$, $b_j(S_j)$ is the next number of the fish sequence, and let $U = U \cup \{S_j\}$, or else go to next step.

Step 3: If $U = M$, algorithm finished, or else go to step 4.

Step 4: If $b_j(S_j)$ is the end of the fish sequence, algorithm finished, or else go to step 2.

2.3.Behavior of Looking for Food

For each fish $\mathbf{X}_i$, select a fish $\mathbf{X}_j$ in its vision range. If $Y_i < Y_j$, fish $\mathbf{X}_i$ moves to $\mathbf{X}_j$, or else reselect a fish. If the number of reselection times is bigger than N_{try}, give up looking for food.

2.4.Clustering Behavior

A fish $\mathbf{X}_i$ searches its neighbors and finds the number of its neighbors nf. If $nf < N\delta$ where $(0 < \delta < 1)$, the center of its neighbors is full of food and the number of its neighbors is not very big. Denoted the center of fish $\mathbf{X}_i$'s neighbors as $\mathbf{X}_{center}$, if $Y_i < Y_{center}$, fish $\mathbf{X}_i$ moves to $\mathbf{X}_{center}$, or else continue looking for food.

2.5.Trailing Behavior

A fish $\mathbf{X}_i$ searches a optimal neighbor $\mathbf{X}_{\max}$. If $Y_i < Y_{\max}$ and the number of fish of $\mathbf{X}_{\max}$' s neighborhood $nf < N\delta$, fish $\mathbf{X}_i$ moves to fish $\mathbf{X}_{\max}$, or else continue looking for food.

2.6.Algorithm Description

Step 1：Initialize the fish.

Step 2：Calculate the fitness value by PFHA.

Step 3：For each fish, execute behaviors such as clustering and trailing, if a behavior leads to a better solution, go to Step 4, or else continue looking for food.

Step 4：Update the optimal solution.

Step 5：Update $d_{i,j}$.

Step 6：If iterative times exceed a given value, algorithm finished, or else go to Step 3.

3.Experimental Results

Assuming that $\mathbf{X} = (x_1, x_2, \cdots, x_n)$, $\mathbf{Y} = (y_1, y_2, \cdots, y_n)$ are two artificial fish, the distance between them is defined as follows:

$$d(\mathbf{X},\mathbf{Y}) = \sum_{i=1}^{n} sign(|x_i - y_i|), \quad (3)$$

where $sign(\bullet)$ is the sign function. $N(\mathbf{X},k)$ denotes the k-neighborhood of $\mathbf{X}$,

$$N(\mathbf{X},k)=\left\{\mathbf{X}'\middle|d(\mathbf{X},\mathbf{X}')<k,\ X'\in D\right\}. \quad (4)$$

The center of a school of artificial fish $\mathbf{X}_1,\mathbf{X}_2,\cdots,\mathbf{X}_n$ is defined as follows:

$$\mathbf{X}_{center}=\left(x_1',x_2',\cdots,x_n'\right), \quad (5)$$

where x_i' is a value make the fomula $\sum_{j=1}^{N}\left(x_i'-x_{ji}\right)^2$ to the minimum. That is a value which is the most common value of the sequence $\left\{x_{1i},x_{2i},\cdots,x_{Ni}\right\}$.

To test the algorithm proposed in this paper, we compare it with method using ant colony optimization (ACO) called IAA in [15]. Our experiments are performed on a PC with Pentium IV 2.4 GHz and 512 Mb RAM, using Matlab 7.0. Table 1 shows the comparable results from 100 experiments for each test data, the first test data are selected from [6], others are selected from CATS (Combinatorial Auction Test Suite, http://cats.stanford.edu). Data from CATS are L5 distribution and no-dominated. It can be seen from table 1 that the AFSA's calculation time is much smaller than IAA, and the best results is much closer to the optimal solution. While the values of m and n is increasing, AFSF performs better than IAA. The results indicates that the AFSA for combinatorial auctions is a fast, robust algorithm.

4.Conclusion

An algorithm based on AFSA for combinatorial auctions is proposed by redefined the initialization process, PFHA to calculate the fitness, and the behaviors such as looking for food, clustering and trailing. Experimental results shows that the improved AFSA is a fast and robust algorithm for combinatorial auctions.

References

[1] M. H. Rothkopf, A. Peke and R. M. Harstad, "Computationally manageable combinational auctions," Management Science*,* vol. 44, Aug. 1998, pp. 1131-1147, doi: 10.1287/MNSC.44.8.1131.

[2] A. Andersson, M. Tenhunen and F. Ygge, "Integer programming for combinatorial auction winner determination " Fourth International Conference on Multi-Agent Systems, IEEE Press, Aug. 2000, pp. 39-46, doi: 10.1109/ICMAS.2000.858429

[3] T. Sandholm, "Approaches to winner determination in combinatorial auctions," Decision Support Systems*,* vol. 28, Mar. 2000, pp. 165-176, doi: 10.1016/S0167-9236(99)00066-4.

[4] M. Schwind, T. Stockheim and F. Rothlauf, "Optimization heuristics for the combinatorial auction problem," Proceedings of the Congress on Evolutionary Computation, IEEE Press, Oct. 2003, pp. 1588-1595, doi: 10.1109/CEC.2003.1299862.

[5] Y. Guo, A. Lin, B. Rodrigues and Y. Zhu, "A Non-exact Approach and Experiment Studies on the Combinatorial Auction Problem," Proceedings of the 38th Annual Hawaii International Conference on System Sciences, IEEE Press, Jan. 2005, pp. 82a-82a, doi: 10.1109/HICSS.2005.34.

[6] P. Chen and D. Wang, "Genetic algorithm for solving winner determination in combinatorial auctions," Journal of Northeastern University (Natural Science), vol. 24, pp. 7-10, Jan. 2003, doi: 10.3778/DBDX.0.2003-01-003.

[7] X. Li, Z. Shao and J. Qian, "An optimizing method based on autonomous animats: Fish-swarm Algorithm," Systems Engineering-Theory & Practice, vol. 2002, pp. 33-39, Nov. 2002, doi: 10.3778/XTLL.0.2002-11-006.

[8] X. Li, F. Lu, G. Tian and J. Qian, "Applications of artificial fish school algorithm in combinatorial optimization problems," Journal of Shandong University (Engineering Science), vol. 34, pp. 65-68, Oct. 2004, doi: 10.3778/SDGY.0.2004-05-015.

[9] G. Huang, "An improved artificial Fish-Swarm algorithm based on Griding Method," Microelectronics & Computer,

vol. 24, pp. 83-87, Jul. 2007, doi: 10.3778/WXYJ.0.2007-07-023.

[10] L. Qu and D. He, "Improved Artificial Fish-School Algorithm and its application for solving numerical derivative," Microelectronics & Computer, vol. 26, pp. 123-126, Mar. 2009, doi: 10.3778/WXYJ.0.2009-05-033.

[11] W. Shu, Z. Yuan and Z. Yin, "Signal MP-based sparse decomposition with Artificial Fish-swarm algorithm," Application Research of Computers, vol. 26, pp. 66-68, Jan. 2009, doi: 10.3778/J.ISSN.0.2009-01-018.

[12] Q. Liu and Y. Liu, "Active shape models based on Artificial Fish-Swarm algorithm," Application Research of Computers, vol. 26, pp. 1935-2937, May 2009, doi: 10.3778/JSYJ.0.2009-05-094.

[13] J. Chen, L. Nie and Y. Zhou, "Machining path planning of robot based on artificial fish-swarm algorithm," Computer Engineering and Applications, vol. 45, pp. 226-228, Aug. 2009, doi: 10.3778/J.ISSN.1002-8331.2009.15.066.

[14] H. Lu, Y. Gao, X. Lu and S. Wu, "Range interval optimization of prestressed anchors based on the least squares support vector machine and artificial fish swarm algorithm," Journal of University of Science and Technology Beijing, vol. 32, pp. 133-138, Jan. 2010, doi: 10.3778/BJKD.0.2010-01-024.

[15] R. Gan, Q. Guo, H. Chang and Y. Yi, "Ant Colony Optimization for Winner Determination in Combinatorial Auctions," Third International Conference on Natural Computation, IEEE Press, Aug. 2007, pp., doi: 10.1109/ICNC.2007.242.

Statistics from 100 Experiments for Each Test Data

m, n	**The optimal solutions**	**Algorithm**	**Average of Time (s)**	**The best result**	**The worst result**	**Average of results**
10, 30	6216.82	AFSA	<1	6216.82	6216.82	6216.82
		IAA	<1	6216.82	6216.82	6216.82
30, 100	127.961	AFSA	10.2	127.961	127.961	127.961
		IAA	11.7	126.264	125.17	124.396
30, 200	152.105	AFSA	27.1	151.656	148.757	146.397
		IAA	38.5	143.954	138.964	134.456
30, 300	166.557	AFSA	74.2	159.137	158.366	157.829
		IAA	86.9	145.641	143.467	139.508
50, 500	274.893	AFSA	224.2	255.061	254.037	249.943
		IAA	244	237.326	231.726	225.882