A PSO-Optimized Nash Equilibrium-Based Task Scheduling Algorithm for Wireless Sensor Network

Jiaye Chen and Wenzhong Guo*

College of Mathematics and Computer Sciences, Fuzhou University, Fuzhou 350108, PR China
cjy_fzu@126.com, guowenzhong@fzu.edu.cn

Abstract. For the dynamic load characteristics of Wireless sensor network, we propose the idea of parallel Coalition and introduce the game theory into the solving of dynamic task allocation problem. In this paper, we design the model of multiple task allocation based on Nash equilibrium, and use runtime of task, Transmission energy consumption and Residual energy to design the utility function of Games. Then we use PSO to find to the point of Nash equilibrium. By using this method, guarantee the task execution effectiveness and improve the utilization rate of networks. Simulation results prove the validity of the algorithm, and can effectively prolong the lifetime of the network.

Keywords: WSN, task allocation, PSO, Game Theory, Nash equilibrium.

1 Introduction

Wireless sensor network (WSN) which includes a large number of sensor nodes is a wireless self-organizing and data-centric network [1]. The biggest drawback of wireless sensor network is that nodes have very limited energy, storage space and computing ability. Task scheduling is a classic problem of extensive research in the field of high-performance computing, and is also the core issues in the area of operating system research. In the operation of parallel and distributed computing systems, In order to effectively use the system resources, an application is usually decomposed into multiple tasks. Systems allocate resources to each task and determine the ordering of tasks execution. Task management is an important module in WSN, and it works together with the mobile management and energy management to monitor energy consumption, dynamic change and the role of task allocation of the sensor nodes in the entire network [1].

Many native and foreign scholars have done much research work on task allocation of WSN during the past several years. Yang et al propose an energy-balanced allocation of a real-time application onto a single-hop cluster of homogeneous sensor nodes connected with multiple wireless channels [2]. An epoch-based application consisting of a set of communicating tasks is considered. Each sensor node is equipped with discrete dynamic voltage scaling (DVS). The time and energy costs of both computation

* Corresponding author.

J. Su et al. (Eds.): ICoC 2013, CCIS 401, pp. 62–73, 2013.

and communication activities are considered. Liu et al propose a method based on elastic neural network to reduce energy consumption under the background of tracking aerial flying targets with the aim of the task allocation of collaborative technique in wireless sensor network [3]. In order to prolong the lifetime, reduce the energy consumption and balance the network load effectively, CHEN et al propose a dynamic Coalition model and its corresponding algorithm of task assignment in wireless sensor network (WSN) [4]. This method describes a cost function according to the execution time, energy consumption and load balance. Particle swarm optimization (PSO) is used to optimize task allocation. And on this basis propose a multi-agent-based architecture for WSNs and construct a mathematical model of dynamic Coalition for the task allocation problem [5].

Since Maynard Smith and Price introduced the ideas of evolutionary into game theory, learning from the analysis method of biological theory of evolution became a new way to calculate Nash equilibrium points and had been obtained abundant outcomes[6~8]. As in [9], the solution of the Nash equilibrium been shown to belong PPAD problem completely. Thomas et al solves the Nash equilibrium by using the genetic algorithms [10]. YI et al built a Grid model of m*n type grid using M/ M/ 1 queue system, and promoted the concept of task scheduling Nash equilibrium among multi-schedulers. The optimal objective of each scheduler is mean complete time per task [11].

This paper also based on the mechanism of dynamic coalition, and PSO was adopted to design a WSNs task allocation algorithm based on game theory. PSO is simple and easy to implement, and with no gradient information and with other advantages, which can be used to solve many complex problems. Our algorithm is able to adapt to the dynamic change of network load and adjust the network running status in time. This paper defines the utility function with the goal of reducing the execution time, reducing transport energy consumption and balancing network energy distribution, and using PSO to obtain the Nash equilibrium of tasks allocation. The results of experiment show its dependability and feasibility. The following will detail description of the problem as well as the specific algorithm implementation.

2 Model of Dynamic Task Allocation

2.1 Parallel Coalition

Coalition formation is a key problem in multi-agent systems. Parallel Coalition [12] is a concurrent generation problem of multiple dynamic coalitions. Parallel Coalition consists of two cases: Crossed Coalition and multi-task Coalition. Crossed Coalition means that an agent to join multiple coalition s or a task can be performed by multiple Coalitions.

Due to the limitations of WSNs such as resource availability and shared communication medium, parallel processing among sensor nodes is a promising solution to provide the demanded computation capacity in WSNs. Considering many points of similarity between WSNs and multi-agent systems, this paper introduces the complicated coalition into WSNs. As shown in Figure 1, a coalition consists of a number of nodes, and tasks are assigned to the selected coalition structure. Using this method we can take full advantage of the core capacity of member nodes, which can lead to finishing the tasks more efficiently and is more suitable for the application environment of WSNs.

2.2 The Concept of Mixed Nash Equilibrium

Game theory is a mathematical decisive approach aiming to solve the problem between competition and cooperation. If there is a competing or collaborative behavior among bodies in the environment, they will tend to adopt some effective strategies to maximize the utility of the individual of group. Generally, game body, strategy and utility are three main elements of game theory. The game body also acts as the player for the game. In general, a game requires at least two players. Besides, the game strategy is the actions of each body which is defined in advance, and each body has their own strategy set. In additional, each player of the game has a utility function to estimate the utility obtained from a certain strategy of the body. Assuming an n-person non-cooperative game, the pure strategy of player p_i is defined as $S^i= (s^i_1, s^i_2, \cdots, s^i_{mi})$, where m_i denotes the number of the pure strategy of p_i. The corresponding mixed strategy of the pure strategy S^i is defined as $x^i= (x^i_1, x^i_2, \cdots, x^i_{mi})$, where x^i meets $x^i_j \geq 0$ and $x^i_1+x^i_2+\cdots+x^i_{mi}=1$. i.e., the player selects the pure strategy s^i_j ($1\leq j\leq m_j$) with probability x^i_j. Then the mixed situation of the game theory can be defined as $X= (x^1, x^2, \cdots, x^n)$.

In this mixed situation, the expected payoff of p_i is defined as follows:

$$u_i(X) = \sum_{j_1=1}^{m_1} \sum_{j_2=1}^{m_2} \cdots \sum_{j_n=1}^{m_n} P_i(s^1_{j_1}, s^2_{j_2}, \ldots, s^n_{j_n}) \bullet x^1_{j_1} \bullet x^2_{j_2} \bullet \ldots \bullet x^n_{j_n} \tag{1}$$

Where P_i $(s^1_{j1}, s^2_{j2}, \cdots, s^n_{jn})$ denotes the gain of player p_i when p_1 select strategy s^1_{j1}, player p_2 select strategy s^2_{j2}, $\cdots$, and player p_n select strategy s^n_{jn}.

Definition 1. If the mixed situation X^* meets $u_i(X^*||x^i)\leq u_i(X^*)$, the mixed situation X^* is the mixed Nash Equilibrium of an n-person non-cooperative game where $X^*||x^i$ denotes that only p_i change its strategy.

Property 1. The mixed situation X^* is the mixed Nash equilibrium of an n-person non-cooperative game if and only if the pure strategy s^i_j meets $u_i(X^*|| s^i_j)\leq u_i(X^*)$.

Proof: Suppose that X^* is the mixed Nash equilibrium. If $u_i(X^*|| s^i_j) \geq u_i(X^*)$, the player p_i will obtain a better gain when it select strategy s^i_j. According to the idea of game theory, Nash equilibrium is the best select of each player, so X^* will not be mixed Nash equilibrium.

2.3 Task Allocation

A wireless sensor network consisting of n heterogeneous wireless sensor nodes distributed in a certain range, and 10% of the node elected as the leader node. The number of Coalition is l, and we define the set of coalitions as $C= (c_1, c_2, \ldots, cl)$, where $l=n*10\%$. A set of independent tasks $T= (t_1, t2, \cdots, t_m)$ arrive at sink node at the same time.

An n-dimensional vector $REQ= (req_1, req_2, \cdots, req_n)$ denotes requirements of tasks, where req_i denotes requirement of task t_i. Through the dynamic topology and routing control, sink node can obtain energy of node and ability of task of node.

In this paper, a matrix $B_k= (b_{ij})_{l\times m}$ is used to record the capacity of different coalition on different tasks, and we defined the execution time as:

$$Time_{ij} = \frac{req_i}{b_{ij}} \tag{2}$$

Where b_{ij} denotes the capacity that i-th coalition executes j-th task, $Time_{ij}$ denotes the time required where j-th task run in i-th coalition.

The energy consumption of wireless sensor networks includes three parts: transmission energy consumption, processing power consumption and access to energy consumption. As the energy of transferring 1 bit data is far greater than the energy of processing 1 bit data, we usually ignore the processing energy consumption and the access to energy consumption. The discussion focused on communication energy consumption in this paper. The minimum transmission energy consumption is $P_{0,trans}$ when the standard distance is d_0 i.e., the distance d_{ij} between i-th node and j-th node determines the energy consumption [13]:

$$P_{i,trans} = \frac{d_{ij}^2}{d_0^2} \times \frac{(4\pi)^2 \beta}{G_t G_r \lambda^2} \times P_{o,trans} \tag{3}$$

Where, G_t denotes emission coefficient, G_r denotes receive coefficient, λ denotes Wireless communication wavelength, β denotes Factor of the energy consumption of the system. As $(4\pi)^2\beta/G_tG_r\lambda^2\times P_{0,trans}$ is a constant, $(d_{ij}/d_0)^2$ is the evaluation index of unit data of transmission energy. To simplify the data, a matrix $COST= (cost_{ij})_{m\times l}$ is used to record transmission energy consumption, $cost_{ij}$ denotes the energy consumption when j-th task transfer data to i-th coalition.

This paper use an n-dimensional vector E to denotes residual energy of coalition. e_i denotes residual energy of i-th coalition. $P(e_i)$ denotes the proportion of residual energy of i-th coalition in the sum of residual energy of entire network.

$$P\left(e_i\right) = e_i \Big/ \sum_{i=1}^{l} e_i \tag{4}$$

In order to prolong the network lifetime, during the process of allocation, we should balance the residual energy of each coalition. The network residual energy average degree is defined as:

$$H = -\sum_{i=1}^{l} P\left(e_i\right) \log^{P(e_i)} \tag{5}$$

Where H denotes the energy entropy of networks. The larger the value of entropy, the more average residual energy distribution, and the longer network lifetime.

3 Our Algorithm

This paper assumes that a coalition is constituted by a number of sensor nodes, and these nodes are mutually closer in distance. Tasks are scheduled on the coalitions, rather than directly on the sensor nodes. Algorithm assigns tasks according to the current situation of networks. With development of energy consumption, the algorithm adaptively adjusts the allocation plan. The condition that the Nash equilibrium scheduling algorithm directly work on coalitions can be established is: a coalition is constituted by several nodes, therefore, a coalition can be considered to be a virtual node which has stronger ability and higher energy. Meanwhile, as mentioned above, both of multi-tasks allocation and solution of Nash equilibrium belong to NP-hard problem, take such an approach can reduce the scale of problem to obtain the solution of the problem quickly and reduce the difficult of experimental simulation. Specific implementation approach of our algorithm is given below.

Definition 2. Three components of game theory in our algorithm:

(1) players of game is s set of non-cooperative tasks, $T=(t_1, t2, ..., t_m)$;

(2) The pure strategy set of players consist of n coalition, coalitions are heterogeneous, and coalitions have own corresponding task ability, transmission consumption and residual energy; The corresponding mixed strategy set of players is $X=(x^1, x^2,\dots, x^n)$, where x^i is called mixed strategy of i-th player;

(3) In game theory, the utility function is an important indicator to measure the gain of players, it defined herein is:

$$u_j^i = w_1 \times nt_j + w_2 \times \cos t_{ij} + w_3 \times e_j \tag{6}$$

Where u_j^i represents utility function which is used to transform multi-target to single target, and denotes the gain that i-th task obtain from j-th coalition. The smaller the value of utility function, the better; nt_j is the sum of $busy_j$ and $Time_{ji}$, $busy_j$ denotes the busy time of j-th coalition; $cost_{ij}$ denotes transmission energy consumption of the i-th player in the j-th coalition; e_j denotes the residual energy of j-th coalition; w_1, w_2 and w_3 denote weight value.

3.1 Nash Equilibrium PSO

In this paper, according to Definition 2, we use PSO to find the point of Nash Equilibrium. And our algorithm is called NEPSO.

We use the floating number matrix to represent the task allocation plan. The utility function is defined to optimize task execution time, energy consumption and energy entropy of network. Then we use utility function to further define the fitness function of PSO.

We use a matrix $x_{m\times l}$ to code the position of a particle:

$$X = (x^1, x^2, \cdots, x^m)^T = \begin{bmatrix} x_1^1 & \dots & x_l^1 \\ \vdots & \ddots & \vdots \\ x_1^m & \dots & x_l^m \end{bmatrix} \tag{7}$$

Where x_j^i denotes the probability that *i*-th task select the *j*-th coalition, and $x_1^i + x_2^i + \cdots + x_l^i = 1$.

For solving the Nash equilibrium of mixed strategies, each task t_i is allocated to some coalitions according to its mixed strategy $x^i = (x_1^i, x_2^i, \cdots, x_l^i)$, In such a case, we need to change the utility function of pure strategies, and the expected utility function is defined as:

$$u_i = (u_1^i, u_2^i, \cdots, u_l^i) \bullet \begin{bmatrix} x_1^i \\ x_2^i \\ \vdots \\ x_l^i \end{bmatrix} = \sum_j^l u_j^i \bullet x_j^i \tag{8}$$

And we also need to update status of coalition after assigned a task:

$$busy_j = busy_j + x_j^i * Time_{ji} \tag{9}$$

$$e_j = e_j - x_j^i * \cos t_{ij} \tag{10}$$

The fitness function of PSO is defined as follows:

$$f(X) = \sum_i \max\{\max\{u_i(X_{i,-i}) - u_i(X^*)\}, 0\} \tag{11}$$

This fitness function is based on the fact: from the point of view of each player, if it change its strategy, the gain that take pure strategy is less than the gain that take mixed strategy, and this player will not want to change its strategy. As shown in equation (10), the value of fitness function of X is zero when X is the best solution X^*. The smaller the value of fitness, the better.

In each time of iteration, the particles update themselves by tracking the two extreme values. One is the optimal solution of each particle, which is called the local optimal solution, denoted by X^i_{lBest}, where N_p denotes the number of particles. The other extreme is the global optimal solution of entire population which is currently found, denoted by X_{gBest}. During the iteration of PSO, the *i*-th particle velocity and position update equation:

$$\begin{aligned} V_k^i(t+1) &= w * V_k^i(t) + c_1 * r_1 * (X_{lBest}^i - X_k^i(t)) \\ &+ c_2 * r_2 * (X_{gBest} - X_k^i(t)) \end{aligned} \tag{12}$$

$$X_k^i(t+1) = V_k^i(t+1) + X_k^i(t) \tag{13}$$

Where $V_k^i(t)$ denotes the speed of the *i*-th particle during the *k*-th iteration, $X_k^i(t)$denotes the position of the *i*-th particle during the *k*-th iteration, X^i_{lBest} denotes the current local optimal solution of *i*-th particle, X_{gBest} denotes the current global optimal solution of entire population. r_1 and r_2 denote the random number between 0-1, c_1 and

c_2 denotes learning factor. w denotes Inertia weight, and it is linearly decreasing weight, and decrease from w_{max} to w_{min}, as shown in equation (13):

$$w = w_{max} - ite \times \frac{w_{max} - w_{\min}}{ite_{max}} \tag{14}$$

Where ite_{max} denotes maximum number of iterations.

Definition 3. if mixed Nash equilibrium solution X meets $\forall i, j$, $x_j^i \geq 0$ and $\sum_j x_j^i = 1$, it is called standardized solution.

If the solution of particles during the iteration of PSO is not the standardized solution, we should deal it with the method shown in equation (14) and (15):

$$\begin{cases} 0, & x_j^i < 0 \\ 1, & x_j^i > 0 \\ x_j^i & 0 \leq x_j^i \leq 1 \end{cases} \tag{15}$$

$$x_j^i = x_j^i \Big/ \sum_j x_j^i \tag{16}$$

3.2 PSO Algorithm Process

Input:
(1) The size of population K, the maximum number ite_{max};
(2) Inertia weight w, maximum weight values w_{max}, minimum weight value w_{min};
(3) Learning factor c_1 and c_2, the value is 2 in our experiments;
(4) Initialize set of tasks $T= (t_1, t2, \cdots, t_m)$, set of tasks requirements $REQ= (req_1, req_2, \cdots, req_n)$, an ability matrix $B = (b_{ij})_{l\times m}$, and the energy consumption matrix $COST= (cost_{ij})_{m\times l}$.

Output:
(1) the best mixed strategy X^*;
(2) Residual energy of each coalition $RE=(re_1, re_2, \ldots, re_l)$;
(3) Busy time of coalitions $BUSY= (busy_1, busy2, \cdots, busy_l)$.

Step1: Initialize the population. Initialize each particle X, each component of the vector x^i is random number between 0-1, then handle x^i according to equation (14) and (15);

Step2: compute $V^i(t+1)$ of i-th particle according to equation (11), then update $X^i(t+1)$ according to equation (12);

Step3: handle $V^i(t+1)$ according to equation (14) and (15);

Step4: compute fitness value of $X^i(t+1)$;

Step4.1: input mixed strategy matrix X, busy time and energy of each coalition, and set of tasks;

Step4.2: for task t_i, compute its executing time and transmission energy consumption in the coalitions;

Step4.3: according to equation (5), compute gain of pure strategy of task t_i in the coalitions.

Step4.4: according to equation (7) and mixed strategy x^i, compute expected gain of task t_i .

Step4.5: update busy time and energy of each coalition according to equation (8) and equation (9);

Step4.6: compare expected gain of task t_i and all pure gain, then update the value of fitness according to equation (10);

Step4.7: if t_i is the last task, then end; else i plus 1 and go to **step4.2**.

Step5: determine whether need to update the local optimal solution or the global optimal solution;

Step6: The number of iterations plus 1;

Step7: Judge whether the number of iterations reaches the upper limit ite_{max}. If $ite=ite_{max}$, then return X_{gBest}, else go to **Step2.**

During the process of computing, we need to handle the three parameters of the utility function (execute time, transmission energy consumption and residual energy). In this paper, the value mapped to the interval [0, 0.5] by using sigmoid function, as shown in equation (16) and equation (17):

$$f(x) = -\frac{1}{1+e^{-x}} + 1 \tag{17}$$

$$f(x) = \frac{1}{1+e^{-x}} - 0.5 \tag{18}$$

$$nt_i = -\frac{1}{1+e^{-\frac{nt_i - nt_{\min}}{nt_{\max} - nt_{\min}}}} + 1 \tag{19}$$

$$\cos t_{ij} = -\frac{1}{1+e^{-\frac{\cos t_{ij} - \cos t_{\min}}{\cos t_{\max} - \cos t_{\min}}}} + 1 \tag{20}$$

$$e_i = -\frac{1}{1+e^{-\frac{e_i - e_{\min}}{e_{\max} - e_{\min}}}} + 1 \tag{21}$$

4 Simulation and Results

Our simulation study is conducted for a WSN of n nodes that are placed uniformly in a rectangular region of 200 by 200 meters, and 10% of the nodes are elected as the leader. The requirements of the subtask are distributed in the range of the interval (2, 6]. In the same situation, the greater the value is, the longer the time of executing this task is. This value also reflects the difficulty of the task processing. The ability of executing task is distributed in the range of the interval (15, 25], the greater the value is, the stronger the ability is. The energy consumption is distributed in the range of the interval (3, 7], the greater the value is, the greater the consumption is. The energy of each node is distributed in the range of the interval (45000, 55000] mj.

Through several experiments, in order to obtain a high-quality solution rapidly in a short period of time, the parameters of PSO are set as follows: maximum number of iterations ite_{max} is 100, the size of population K is 50, w_{max} is 0.9, w_{min} is 0.5, c_1 and c_2 is 2, w_1 is 1, w_2 is 1, w_3 is 3.

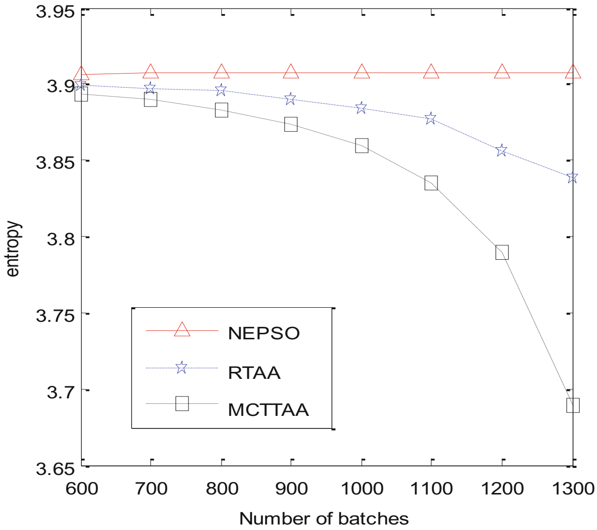

Fig. 1. The entropy of different numbers of batches

As shown in Figure 1, here are a set of experiments to observe the performance of wireless sensor networks under different task batch. In the respect of balancing networks energy to improve networks lifecycle, compare to MCTTAA and RTAA, NEPSO shows good results. The Energy Entropy is keeping at about 3.9. From this figure, we can know that three algorithm can let the network has a good entropy when the batches of task is small, especially at the interval [600, 900]. However, with the increasing of the batches of task, entropy of MCTTAA declining much faster than the others. Similarly, although the RTAA let network energy entropy still maintaining at a good level, but compared to the NEPSO algorithm, it is more poor.

Figure 2 and Figure 3 are the compare of execution time. Due to MCTTAA is based on the shortest completion time, whether the average execution time or minimum execution time, it shows a very good performance. RTAA and NEPSO is worse. As shown in Figure 8, the average execution time of NEPSO and RTAA is almost the same, and their corresponding curves are almost overlapping. And on the minimum execution time, as shown in Figure 9, NEPSO after MCTTAA is superior to RTAA.

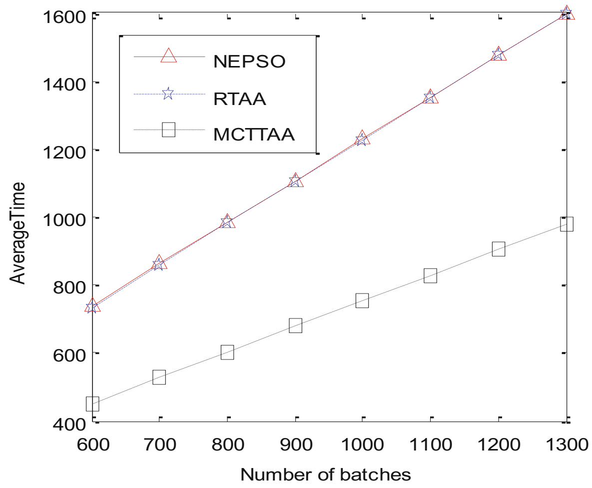

Fig. 2. The average of execution time of different numbers of batches

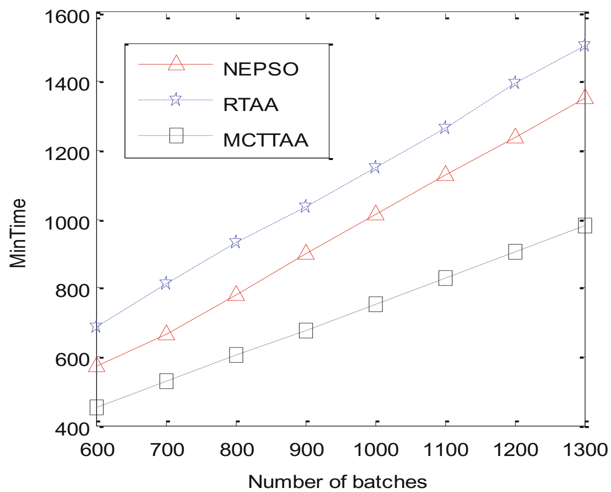

Fig. 3. The minimum of execution time of different numbers of batches

In this experiment, the maximum residual energy in coalition is set to be 55000*mj*.When an coalition's residual energy is less than 5% of the maximum residual energy, namely residual energy is less than 2750*mj*, the network will be failure. As shown in Fig.4, under different numbers of alliance, the batches of task executed by NEPSO are the most. When the number of coalition is small, the disparity among the three algorithms is not obvious, but with the increase of coalitions, it can obviously see that performance of NEPSO in improving the network life cycle is excellent, RTAA and MCTTAA are much poor, especially MCTTAA.

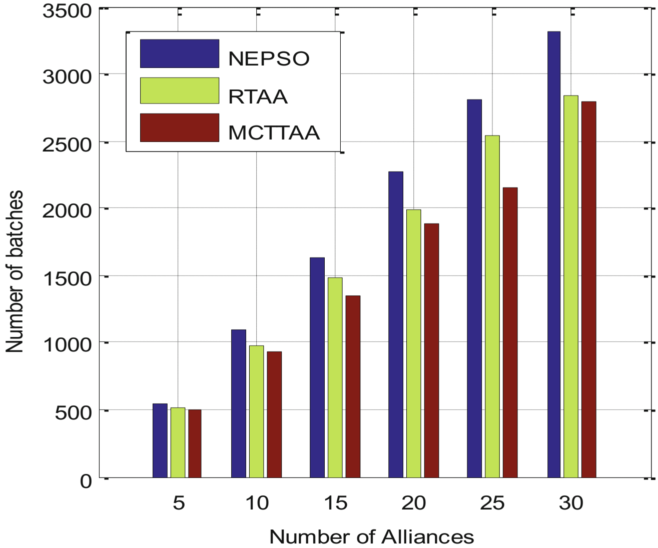

Fig. 4. The life cycle of different numbers of coalitions

5 Conclusion

For certain characteristics of wireless sensor networks, based on dynamic coalition mechanism, this paper propose a task dynamic allocation algorithm using game theory. The proposed algorithm designs a strategy to solve the Nash equilibrium with PSO algorithm. Simulation results show that the adaptive algorithm constructed in this paper is effective. It can obtain a satisfactory solution in a short time and ensure the execution time while effectively extend the lifetime of network. Further research work will focus on the fault-tolerant mechanism, namely, building a tasks adaptive allocation algorithm with fault-tolerant mechanism in WSN.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China under Grant No.61103175 and No. 61103194, the Key Project of Chinese Ministry of Education under Grant No.212086, the Technology Innovation Platform Project of Fujian Province under Grant No.2009J1007, the Key Project Development Foundation of Education Committee of Fujian province under Grand No. JA11011, and the Fujian Province High School Science Fund for Distinguished Young Scholars under Grand No.JA12016.

References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., et al.: Wireless sensor networks: a survey. Computer Networks 38, 393–422 (2002)
2. Yu, Y., Viktor, K.P.: Energy-balanced task allocation for collaborative processing in wireless sensor networks. Mobile Networks and Applications 10(12), 115–131 (2005)

3. Liu, M., Li, H.H., Shen, Y.: Research on task allocation technique for aerial target tracking based on wireless sensor network. Journal of Astronautics 28(4), 960–965 (2007)
4. Chen, G.L., Guo, W.Z., Chen, Y.Z.: Research on dynamic alliance of task allocation and its algorithm in wireless sensor network. Journal on Communication 30(11), 48–55 (2009)
5. Guo, W.Z., Xiong, N.X., Chao, H.C., et al.: Design and Analysis of Self-Adapted Task Scheduling Strategies in Wireless Sensor Networks. Sensors 11(7), 6533–6554 (2011)
6. Maynard, S.J.: Evolution and the Theory of Games. Cambridge University Press, Cambridge (1982)
7. Thomas, R.: Genetic Algorithm Learning and Evolutionary Games. Journal of Economic Dynamics & Control 25, 1019–1037 (2001)
8. Chen, S.J., Sun, Y.G., Wu, Z.G.: A Genetic Algorithm to Acquire the Nash Equilibrium. System Engineering 19(5), 67–70 (2001)
9. Daskalakis, C., Goldberg, P.W., Papadimitriou, C.H.: The complexity of computing Nash equilibrium. In: STOC 2006, pp. 71–78. ACM press, Seattle (2005)
10. Thomas, R.: Genetic algorithm learning and evolutionary games. Journal of Economic Dynamics & Control 25(6), 1019–1037 (2001)
11. Kan, Y., Wang, R.C.: Nash Equilibrium Based Task Scheduling Algorithm of Multi-schedulers in Grid Computing. Acta Electronica Sinica 37(2), 329–333 (2009)
12. Zhang, G.F., Jiang, J.G., Xia, N., Su, Z.P.: Solutions of complicated coalition generation based on discrete particle swarm optimization. Acta Electronica Sinica 35(2), 323–327 (2007)
13. Wu, Q., Rao, N.S., Barhen, J.: On computing mobile agent routes for data fusion in distributed sensor networks. IEEE Trans. on Knowledge and Data Engineering 16(6), 740–753 (2004)
14. Garey, M.R., Johnson, D.S.: “Strong” NP-Completeness Results: Motivation, Examples, and Implications. Journal of the Association for Computing Machinery 25(3), 499–508 (1978)
15. Mezura-Montes, E., Coello, C.A.C.: A simple multi-membered evolution strategy to save constrained optimization problems. IEEE Transactions on Evolutionary Computation 9(1), 1–17 (2005)