Multi-objective Peer-to-Peer Neighbor-Selection Strategy Using Genetic Algorithm

Ajith Abraham[1,3], Benxian Yue[2], Chenjing Xian[3],
Hongbo Liu[2,3], and Millie Pant[4]

[1] Centre for Quantifiable Quality of Service in Communication Systems,
Norwegian University of Science and Technology, N-7491 Trondheim, Norway
ajith.abraham@ieee.org
[2] Department of Computer, Dalian University of Technology, 116023 Dalian, China
yuebenxian@vip.sina.com, lhb@dlut.edu.cn
[3] School of Computer Science, Dalian Maritime University, 116026 Dalian, China
Xcj2003@newmail.dlmu.edu.cn
[4] Department of Paper Technology, Indian Institute of Technology - Roorkee,
Saharanpur 247 001, India
millifpt@iitr.ernet.in

Abstract. Peer-to-peer (P2P) topology has significant influence on the performance, search efficiency and functionality, and scalability of the application. In this paper, we present a Genetic Agorithm (GA) approach to the problem of multi-objective Neighbor Selection (NS) in P2P Networks. The encoding representation is from the upper half of the peer-connection matrix through the undirected graph, which reduces the search space dimension. Experiment results indicate that GA usually could obtain better results than Particle Swarm Optimization (PSO).

1 Introduction

Peer-to-peer computing has attracted great interest and attention of the computing industry and gained popularity among computer users and their networked virtual communities [1]. It is no longer just used for sharing music files over the Internet. Many P2P systems have already been built for some new purposes and are being used. An increasing number of P2P systems are used in corporate networks or for public welfare (e.g. providing processing power to fight cancer) [2]. P2P comprises peers and the connections between these peers. These connections may be directed, may have different weights and are comparable to a graph with nodes and vertices connecting these nodes. Defining how these nodes are connected affects many properties of an architecture that is based on a P2P topology, which significantly influences the performance, search efficiency and functionality, and scalability of a system. A common difficulty in the current P2P systems is caused by the dynamic membership of peer hosts. This results in a constant reorganization of the topology [3], [4], [5], [6].

The simplest neighbor selection strategy would be to select a node at random from the candidate nodes. Kurmanowytsch et al. [7] developed the P2P

S. Aluru et al. (Eds.): HiPC 2007, LNCS 4873, pp. 443–451, 2007.

middleware systems to provide an abstraction between the P2P topology and the applications that are built on top of it. These middleware systems offer higher-level services such as distributed P2P search and support for direct communication among peers. The systems often provide a pre-defined topology that is suitable for a certain task (e.g., for exchanging files).

Koulouris et al. [8] presented a framework and an implementation technique for a flexible management of peer-to-peer overlays. The framework provides means for self-organization to yield an enhanced flexibility in instantiating control architectures in dynamic environments, which is regarded as being essential for P2P services to access, routing, topology forming, and application layer resource management. In these P2P applications, a central tracker decides about which peer becomes a neighbor to which other peers.

Koo et al. [9] investigated the neighbor-selection process in the P2P networks, and proposed an efficient single objective neighbor-selection strategy based on Genetic Algorithm (GA). Sun et al. [10] proposed a PSO algorithm for neighbor selection in P2P networks. In this paper, we explore the multi-objective neighbor-selection problem based on GA for P2P Networks.

This paper is organized as follows. We formulate the problem in Section 2. The proposed approach based on genetic algorithm is presented in Section 3. In Section 4, experiment results and discussions are provided in detail, followed by some conclusions in Section 5.

2 Neighbor-Selection Problem in P2P Networks

Kooa et al. modeled the neighborhood selection problem using an undirected graph and attempted to determine the connections between the peers [9], [11]. Given a fixed number of N peers, we use a graph $G = (V, E)$ to denote an overlay network, where the set of vertices $V = \{v_1, \cdots, v_N\}$ represents the N peers and the set of edges $E = \{e_{ij} \in \{0, 1\}, i, j = 1, \cdots, N\}$ represents their connectivities : $e_{ij} = 1$ if peers i and j are connected, and $e_{ij} = 0$ otherwise. For an undirected graph, it is required that $e_{ij} = e_{ji}$ for all $i \neq j$, and $e_{ij} = 0$ when $i = j$. Let C be the entire collection of content fragments, and we denote $\{c_i \subseteq C, i = 1, \cdots, N\}$ to be the collection of the content fragments each peer i has. We further assume that each peer i will be connected to a maximum of d_i neighbors, where $d_i < N$. The disjointness of contents from peer i to peer j is denoted by $c_i \setminus c_j$, which can be calculated as:

$$c_i \setminus c_j = c_i - (c_i \cap c_j). \tag{1}$$

where $\setminus$ denotes the exclusion operator, and $\cap$ intersection operation on sets. This disjointness can be interpreted as the collection of content fragments that peer i has but peer j does not. In other words, it denotes the fragments that peer i can upload to peer j. Moreover, the disjointness operation is not commutative, i.e., $c_i \setminus c_j \neq c_j \setminus c_i$. We also denote $|c_i \setminus c_j|$ to be the cardinality of $c_i \setminus c_j$, which is the number of content fragments peer i can contribute to peer j. In order to maximize the disjointness of content, we want to maximize the number of content fragments each peer can contribute to its neighbors by determining the connections e_{ij}'s. Define ϵ_{ij}'s to be sets such that $\epsilon_{ij} = C$ if $e_{ij} = 1$, and $\epsilon_{ij} = \emptyset$ (null set) otherwise. Therefore the neighbor selection can be formulated as the following optimization problem:

$$\max_{E} \sum_{j=1}^{N} \left| \bigcup_{i=1}^{N} (c_i \setminus c_j) \cap \epsilon_{ij} \right| \tag{2}$$

It is desirable to select peers with the most mutually disjoint collection of content fragments as neighbors. However, downloading the file fragments between each peer pair would consume away the bandwidth and connect cost, etc. τ_{ij} describes the cost coefficient between peer i and j. The performance of the whole system would be emphasized. The neighbor selection strategy is expected not only to assure maximum content availability but also to minimize the downloading cost to improve the overall throughput of the system. So the objectives are summarized as follows:

$$f_1(x) = \max_{E} \sum_{j=1}^{N} \left| \bigcup_{i=1}^{N} (c_i \setminus c_j) \cap \epsilon_{ij} \right| \tag{3}$$

$$f_2(x) = \min_{E} \sum_{j=1}^{N} \sum_{i=1}^{N} \tau_{ij} |(c_i \setminus c_j)||\epsilon_{ij}| \tag{4}$$

Subject to

$$\sum_{j=1}^{N} e_{ij} \leq d_i \text{ for all } i \tag{5}$$

3 Genetic Algorithm for Multi-objective Neighbor Selection

Multi-objective genetic algoritm has been a very popular multiobjective technique, and it normally exhibits a very good overall performance. Many multi-objective optimization techniques using evolutionary algorithms have been proposed in recent years [12], [13], [14]. Given a P2P state $S = (N, C, M, F)$, in which N is the number of peers, C is the entire collection of content fragments, M is the maximum number of the peers which each peer can connect steadily in the session, F is to goal the number of swap fragments, i.e. to maximize equation (3) and minimize equation (4) with the constraint in equation (5). To apply the genetic algorithm successfully for the NS problem, one of the key issues is the mapping of the problem solution into the search space, which directly affects its feasibility and performance. The neighbor topology in P2P networks is an undirected graph, i.e. $e_{ij} = e_{ji}$ for all $i \neq j$. We set up a search space of D dimension as $N * (N - 1)/2$. Accordingly, each individual is represented as a binary bit string of length D. Each dimension maps one undirected connection.

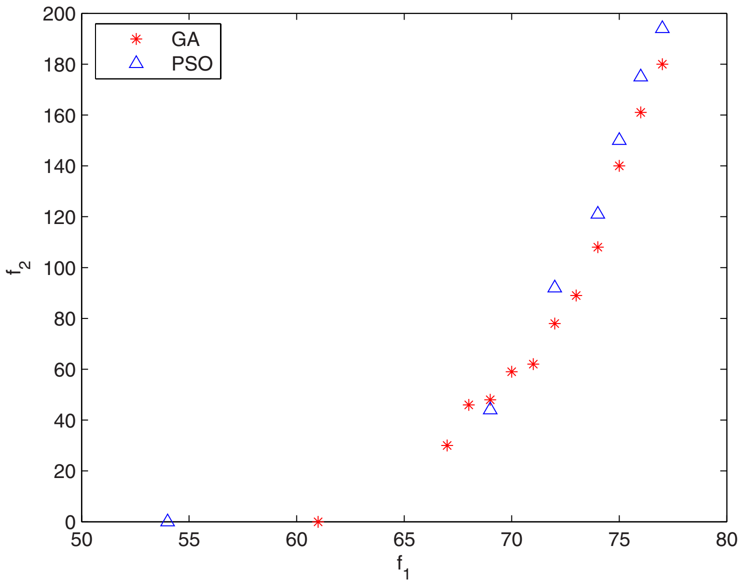

Fig. 1. Performance for the NS $(6, 60, 3)$

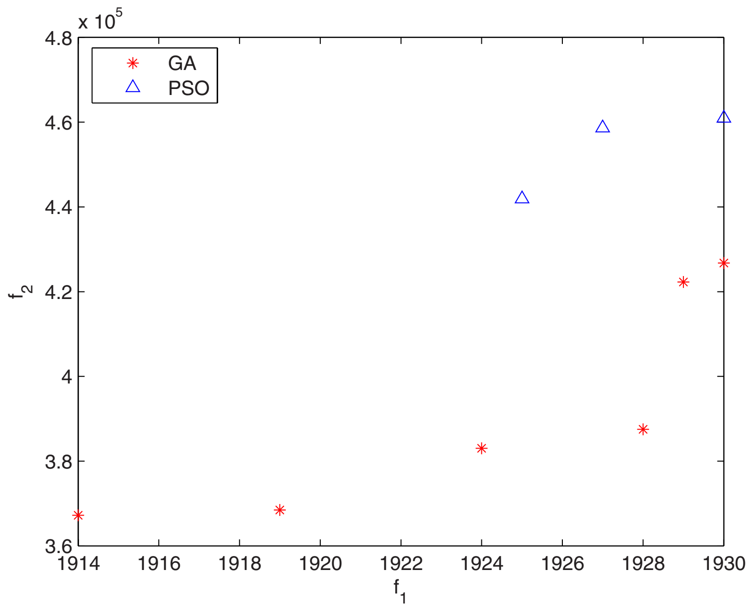

Fig. 2. Performance for the NS $(25, 300, 12)$

The domain for each dimension is limited to 0 or 1. The binary string has a priority levels according to the order of peers. The sequence of the peers will be not changed during the iteration. It indicates the potential connection state. The pseudo-code for our P2P neighbor selection method is illustrated in Algorithm 1.

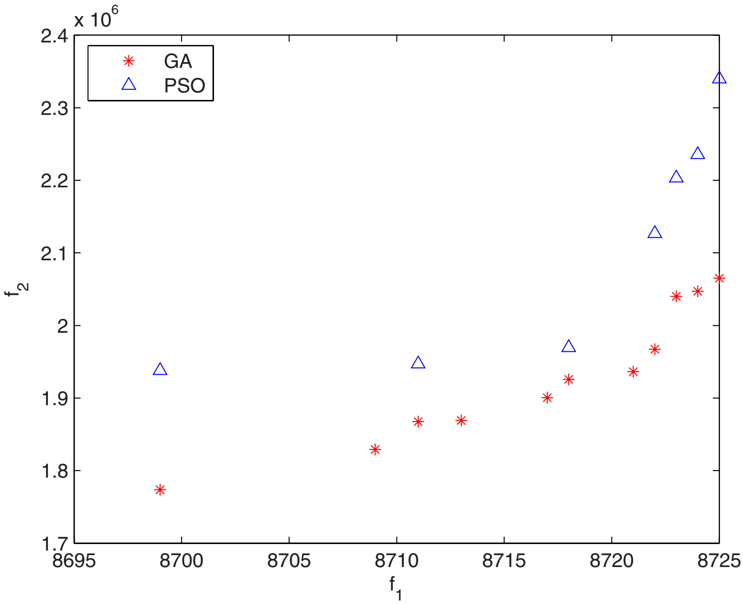

Fig. 3. Performance for the NS $(25, 1400, 12)$

4 Algorithm Performance Demonstration

To illustrate the effectiveness and performance of our algorithm, we demonstrate an execution trace of the algorithm for the NS problem. A file of size 7 MB is divided into 14 fragments (512 KB each) to distribute, 6 peers download from the P2P networks, and the connecting maximum number of each peer is 3, which is represented as $(6, 14, 3)$ problem. In some session, the state of distributed file fragments is as follows:

$$\begin{bmatrix} 1 & 0 & 0 & 4 & 0 & 6 & 7 & 8 & 0 & 10 & 0 & 12 & 0 & 14 \\ 0 & 0 & 0 & 4 & 5 & 0 & 7 & 0 & 9 & 0 & 11 & 0 & 13 & 0 \\ 0 & 2 & 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 & 11 & 12 & 0 & 14 \\ 0 & 2 & 3 & 4 & 0 & 6 & 0 & 0 & 0 & 0 & 11 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 7 & 8 & 0 & 10 & 0 & 12 & 0 & 14 \\ 1 & 2 & 0 & 0 & 5 & 0 & 0 & 0 & 9 & 10 & 11 & 0 & 13 & 14 \end{bmatrix}$$

The cost matrix is as follows:

$$\begin{bmatrix} 0 & 5 & 2 & 4 & 1 & 0 \\ 5 & 0 & 3 & 0 & 2 & 2 \\ 2 & 3 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 5 & 2 \\ 1 & 2 & 0 & 5 & 0 & 10 \\ 0 & 2 & 0 & 2 & 10 & 0 \end{bmatrix}$$

Algorithm 1. Neighbor Selection Algorithm Based on GA

Initialize the population, and other parameters.
While (the end criterion is not met) do
  Evaluate();
    for $i = 1$ to N
      for $j = 1$ to N
        if $j == i$, $e_{ij} = 0$;
          else if $j < i$, $a = j; b = i$;
            else if $j > i$, $a = i; b = j$;
          $e_{ij} = p_{[a*N+b-(a+1)*(a+2)/2]}$;
        If $e_{ij} = 1$, calculate $c_i \setminus c_j$;
          Calculate $f_2 = f_2 + \tau_{ij}|(c_i \setminus c_j)|$;
      Next j
      calculate $f_1 = f_1 + \left|\bigcup_{i=1}^{N}(c_i \setminus c_j) \cap \epsilon_{ij}\right|$;
    Next i
  Rank();
  If $nondomCtr > Maxarchivesize$, maintenance-archive();
  Generate-new-pop();
    Crossover();
    Mutation();
  $t++$;
  If rank $==$ 1 output the fitness;
End While.

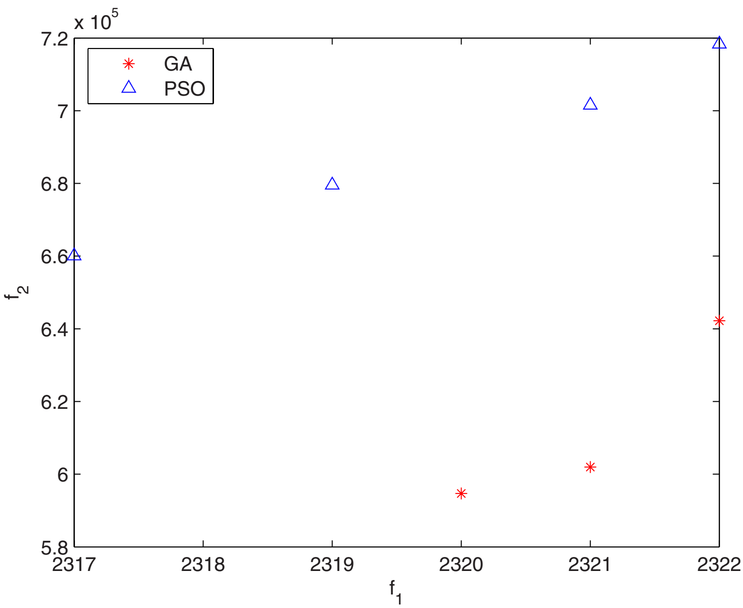

Fig. 4. Performance for the NS $(30, 300, 15)$

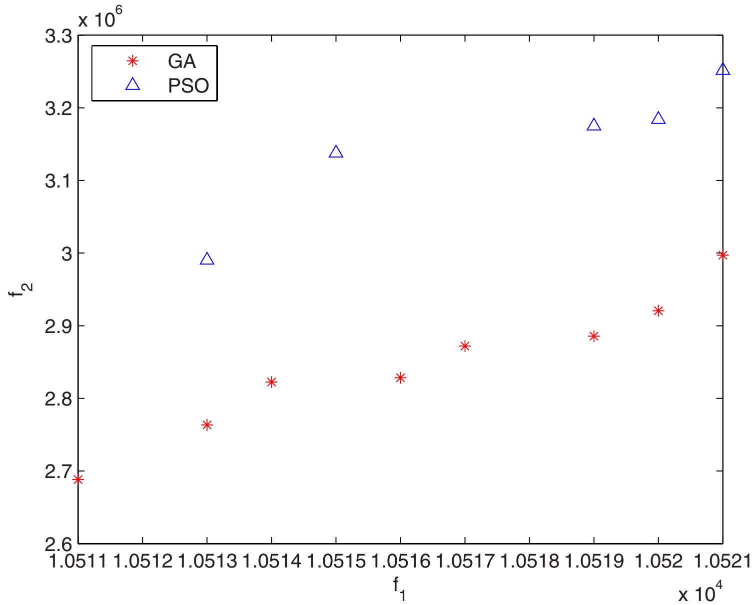

Fig. 5. Performance for the NS $(30, 1400, 15)$

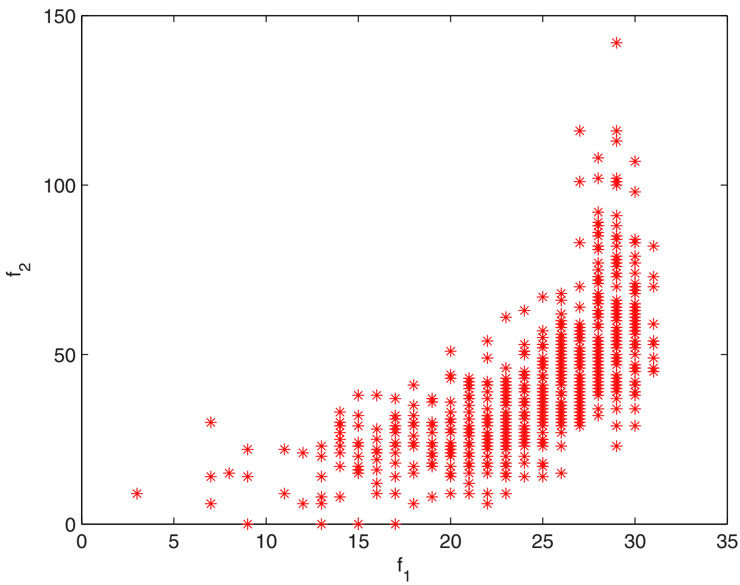

Fig. 6. Performance for the NS $(6, 60, 3)$

The performance output is illustrated in Figure 6 by the proposed algorithm. We also tested other five representative instances (problem (6,60,3), problem (25,300,12), problem (25,1400,12), problem (30,300,15), problem (30,1400,15)) further. In our experiments, the algorithms used for comparison were GA (Genetic Algorithm) and PSO (Particle Swarm Optimization). The GA and PSO algorithms share many similarities [15], [16], [17].

In GA, a population of candidate solutions (for the optimization task to be solved) is initialized. New solutions are created by applying reproduction operators (mutation and crossover). The fitness (how good the solutions are) of the resulting solutions are evaluated and suitable selection strategy is then applied to determine which solutions will be maintained to the next generation. PSO algorithm is inspired by social behavior patterns of organisms that live and interact within large groups. It incorporates swarming behaviors observed in flocks of birds, schools of fish, or swarms of bees, and even human social behavior. The PSO/GA algorithms were repeated 3 times with different random seeds. Each trial had a fixed number of 200 iterations. Other specific parameter settings of the algorithms are described in Table 1. The average fitness values of the best (rank = 1) solutions throughout the optimization run were recorded.

Figures 1, 2, 3, 4 and 5 illustrate the GA/PSO performance during the search processes for the NS problem. As evident, GA usually obtained better results than PSO.

Table 1. Parameter settings for the algorithms.

Algorithm	Parameter name	value
	size of the population	$left\ even\ number(10 + 2sqrt(D))$
GA	Probability of crossover	0.8
	Probability of mutation	0.08
	Swarm size	$left\ even\ number(10 + 2sqrt(D))$
	Self coefficient c_1	2
PSO	Social coefficient c_2	2
	Inertia weight w	0.9
	Clamping Coefficient ρ	0.5

5 Conclusions

In this paper, we investigated the problem of multi-objective neighbor selection in peer-to-peer networks using genetic algorithm. In the proposed strategy, the solution encoding was done from the upper half matrix of the peer connection through the undirected graph, which reduces the dimension of the search space. We evaluated the performance of the genetic algorithm with particle swarm optimization algorithm. Empirical results indicate that GA usually obtain better results than PSO. The proposed algorithm could be an ideal approach for solving the multi-objective NS problem.

Our future work is targeted to test more complicated instances in an online environment of P2P networks and involve more intelligent/heuristics approaches.

References

1. Kwok, S.: P2P searching trends: 2002-2004. Information Processing and Management 42, 237–247 (2006)
2. Idris, T., Altmann, J.: A Market-managed topology formation algorithm for peer-to-peer file sharing networks. In: Stiller, B., Reichl, P., Tuffin, B. (eds.) ICQT 2006. LNCS, vol. 4033, pp. 61–77. Springer, Heidelberg (2006)
3. Surana, S., Godfrey, B., Lakshminarayanan, K., Karp, R., Stoica, I.: Load balancing in dynamic structured peer-to-peer systems. Performance Evaluation 63, 217–240 (2006)
4. Duan, H., Lu, X., Tang, H., Zhou, X., Zhao, Z.: Proximity Neighbor Selection in Structured P2P Network. In: Proceedings of the Sixth IEEE International Conference on Computer and Information Technology (CIT 2006), vol. 52 (2006)
5. Schollmeier, R.: A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications. In: Proceedings of the First International August Conference on Peer-to-Peer Computing, pp. 101–102 (2001)
6. Ghosal, D., Poon, B.K., Kong, K.: P2P Contracts: A Framework for Resource and Service Exchange. Future Generation Computer Systems 21, 333–347 (2005)
7. Kurmanowytsch, R., Kirda, E., Kerer, C., Dustdar, S.: OMNIX: A topology-independent P2P middleware. In: Eder, J., Missikoff, M. (eds.) CAiSE 2003. LNCS, vol. 2681, Springer, Heidelberg (2003)
8. Koulouris, T., Henjes, R., Tutschku, K., de Meer, H.: Implementation of adaptive control for P2P overlays. In: Wakamiya, N., Solarski, M., Sterbenz, J.P.G. (eds.) IWAN 2003. LNCS, vol. 2982, pp. 292–306. Springer, Heidelberg (2004)
9. Koo, S.G.M., Kannan, K., Lee, C.S.G.: On neighbor-selection strategy in hybrid peer-to-peer networks. Future Generation Computer Systems 22, 732–741 (2006)
10. Sun, S., Abraham, A., Zhang, G., Liu, H.: A Particle Swarm Optimization Algorithm for Neighbor Selection in Peer-to-Peer Networks. In: 6th International Conference on Computer Information Systems and Industrial Management Applications, pp. 166–172. IEEE Computer Society Press, Los Alamitos (2007)
11. Kessler, J., Rasheed, K., Arpinar, I.B.: Using genetic algorithms to reorganize superpeer structure in peer to peer networks. Applied Intelligence 26, 35–52 (2007)
12. Abraham, A.: Evolutionary Computation. In: Sydenham, P., Thorn, R. (eds.) Handbook for Measurement Systems Design, pp. 920–931. John Wiley and Sons Ltd, London (2005)
13. Abraham, A., Jain, L.: Evolutionary Multiobjective Optimization. In: Abraham, A., Jain, L.C., Goldberg, R. (eds.) Evolutionary Multiobjective Optimization: Theoretical Advances and Applications, ch. 1, pp. 1–9. Springer, London (2005)
14. Srinivas, N., Deb, K.: Multiobjective optimization using nondominated sorting genetic algorithms. Evolutionary Computation 2(3), 221–248 (1994)
15. Clerc, M.: Particle Swarm Optimization. ISTE Publishing Company, London (2006)
16. Abraham, A., Guo, H., Liu, H.: Swarm Intelligence: Foundations, Perspectives and Applications. In: Nedjah, N., Mourelle, L. (eds.) Studies in Computational Intelligence, pp. 3–25. Springer, Heidelberg (2006)
17. Raquel, C., Naval, P.: An Effective Use of Crowding Distance in Multiobjective Particle Swarm Optimization. In: Proceedings of Genetic and Evolutionary Computation Conference (GECCO 2005), pp. 25–29. ACM, New York (2005)