A PBIL for Load Balancing in Network Coding based Multicasting

Huanlai Xing[1,]*, Ying Xu[2], Rong Qu[3], Lexi Xu[4]

[1] School of Information Science and Technology,
Southwest Jiaotong University, Chengdu, P. R. China
hxx@home.swjtu.edu.cn
[2] College of Computer Science and Electronic Engineering,
Hunan University, Changsha, P. R. China
[3] School of Computer Science,
University of Nottingham, Nottingham, UK
[4] Department of Network Optimisation and Management,
China Unicom Network Technology Research Institute, Beijing, P. R. China

Abstract. One of the most important issues in multicast is how to achieve a balanced traffic load within a communications network. In the context of multicast with network coding, this paper formulates a load balancing optimization problem and proposes a modified population based incremental learning (PBIL) algorithm for tackling it. A novel probability vector update scheme is developed to enhance the global exploration of the stochastic search by introducing extra flexibility when guiding the search towards promising areas in the search space. Experimental results demonstrate that the proposed PBIL outperforms a number of the state-of-the-art evolutionary algorithms in terms of the quality of the best solution obtained.

Keywords: load balancing; multicast; network coding; population based incremental learning

1 Introduction

With the popularity of the Internet, network traffic has been dominated by a dramatically increasing number of multimedia applications, e.g. online games, IPTV, VoD, remote education, and video conferencing. It has been reported that over 90% of the Internet traffic comes from multimedia. Multicast is one of the efficient technologies developed for supporting one-to-many multimedia applications with stringent quality-of-service (QoS). Therefore, this technology has drawn significant amount of research attention from both academia and industry [1]. Unfortunately, multicast with store-and-forward forwarding cannot guarantee a theoretical maximal throughput is always obtained. Network coding allows intermediate nodes to perform mathematical operations to incoming information if necessary. When incorporated into multicast, it can always have the theoretical maximal throughput achieved and hence becomes an ideal technique for point-to-multipoint data transmission [2].

Load balancing is an issue inherited from the adaption of network resource allocation. Network service providers are eager to make use of the network infrastructure as fully as possible so as to accommodate more users with different QoS guarantees. Undoubtedly, a more balanced traffic load helps make use of the remaining network resources functioning more efficiently. Hence, load balancing has been a hot spot in the field of communications for many years [3]. There are some research efforts dedicated to the load balancing in network coding based multicast (NCM). A NCM algorithm was presented and compared with two traditional multicast routing algorithms with respect to the achievable throughput and load balancing [4]. A reliable data dissemination protocol, which adapts network coding for decreasing the broadcast traffic in code updates, was developed in [5]. In [6], the problem of exploiting the abilities of next generation terminals in satellite systems with network coding was investigated. In multi-hop wireless networks, a flexible energy-efficient multicast routing algorithm with network coding was put forward [7]. However, all work above assumes that coding recombination has to be executed at all coding-possible nodes within a network, which would consume serious network computational and buffering resources as coding incurs expensive computational overhead [8-10]. Since coding might only be necessarily performed at a limited number of coding-possible nodes, it is of vital importance to consider the load balancing issue in NCM, where coding is to be performed when needed. However, this issue has received little attention.

Population based incremental learning (PBIL) is an estimation of distribution algorithm, incorporating competitive learning concept into genetic algorithm (GA). PBIL builds a probability model and evolves it to lead the search towards promising areas in the search space. Due to the simplicity and efficiency, the algorithm has applied to a wide range of optimization problems, including the stabilizer design problem in power system [11], the dynamic optimization problem [12], the robot soccer system optimization [13], the antenna design problem [14], the network coding resource minimization problem [15,16], etc.

In this paper, a load balancing optimization problem in the context of NCM is formulated, where coding is performed only when necessary. A modified PBIL with a new probability vector update scheme is adopted to optimize the problem above. The new scheme maintains a set of best-so-far samples obtained during the evolution. At each generation, a proportion of samples in the set are randomly selected and used to update the probability vector, which to a certain extent, helps improve the global exploration. Performance comparisons show that the proposed PBIL is superior to a number of state-of-the-art evolutionary algorithms (EAs) with respect to the solution quality.

2 Problem Formulation

We represent a communications network by a directed graph $G = (V, E)$, where V and E are sets of nodes and links, respectively. The number of links in E is denoted by $|E|$.

Assume every link in G is numbered and let $e_i \in E$ denote link i, where e_i is associated with a maximum bandwidth B_i^{max} and a currently consumed bandwidth B_i^{csmd}, where $B_i^{max} \geq B_i^{csmd}$. There is a source node $s \in V$, a set of receivers $T = \{t_1,\ldots, t_d\}$, $t_k \in V$, $k = 1, \ldots, d$, and an expected bandwidth $R_{s\rightarrow T}$ from s to each receiver in T [8,9].

We call an intermediate node in G as a *merging* node if it is non-source, non-receiver and has multiple incoming links. Only merging nodes can perform coding operations. So, all coding-possible nodes in G are merging nodes. Given a NCM request, the task is to find a connected subgraph (i.e. sub-network) in G to deliver NCM data traffic [9]. We refer to such subgraph as NCM subgraph and denote it by $G_{s\rightarrow T}$. A NCM subgraph consists of multiple paths, with each path originating from source s and terminating at a receiver. We refer to *link-disjoint* paths as paths which do not have any common link. More details can be found in [9].

In this paper, identical bandwidth consumption, denoted by $B_{s\rightarrow T}$, incurs in each link occupied by the NCM subgraph. So, R link-disjoint paths to the same receiver will incur $R\cdot B_{s\rightarrow T}$ bandwidth consumption, where R is an integer. Let ω_i be the bandwidth utilization ratio in $e_i \in E$, Φ be the number of link-disjoint paths to each receiver in $G_{s\rightarrow T}$, and $\Omega_z(s\rightarrow t_k)$ be path z from s to t_k in $G_{s\rightarrow T}$, $z=1, \ldots, \Phi$, respectively. Let $\rho_z(s\rightarrow t_k)$ and $r(s\rightarrow t_k)$ denote the link set of $\Omega_z(s\rightarrow t_k)$ and the achievable bandwidth from s to $t_k \in T$ in $G_{s\rightarrow T}$, respectively.

In a communications network, each link is associated with a bandwidth utilization ratio (BUR) which reflects the percentage of how much bandwidth has been used. The task of this paper is to find a NCM subgraph in G, with the average BUR minimized and a number of constraints met. A smaller average BUR implies a more balanced traffic load in G.

Minimize: $$\overline{\omega} \tag{1}$$

where,

$$\overline{\omega} = \left(\sum_{i\in|E|} \omega_i\right)/|E| \tag{2}$$

$$\omega_i = \left(B_{s\rightarrow T} \cdot c_i + B_i^{csmd}\right)/B_i^{max} \tag{3}$$

$$c_i = \begin{cases} 1, & e_i \in G_{s\rightarrow T} \\ 0, & otherwise \end{cases} \tag{4}$$

Subject to:

$$B_{s\rightarrow T} + B_i^{csmd} \leq B_i^{max}, \quad \forall i = \{1, \ldots, |E|\} \tag{5}$$

$$r(s\rightarrow t_k) = \Phi \cdot B_{s\rightarrow T}, \quad \forall t_k \in T \tag{6}$$

$$\rho_m(s\rightarrow t_k) \cap \rho_n(s\rightarrow t_k) = \emptyset, \quad \forall m,n \in \{1, \ldots, \Phi\}, m \neq n \tag{7}$$

The objective of the problem concerned is to minimize the average BUR, which is shown in Eq. (1). Eq. (2) defines the average BUR. The BUR value associated with each link is calculated based on Eq. (3). Eq. (4) defines the coefficient c_i. Constraint (5) explains the bandwidth constraint. The achievable bandwidth $r(s \rightarrow t_k)$ is Φ times larger than $B_{s \rightarrow T}$ because Φ link-disjoint paths from the source to each receiver are to be constructed, which is shown in Constraint (6). Constraint (7) reflects that any two paths in $G_{s \rightarrow T}$, cannot have common link, as long as they terminate at the same receiver.

3 The Proposed PBIL

This section first introduces the new probability vector (PV) update scheme and then describes the overall procedure of the proposed PBIL in detail.

3.1 New PV update scheme

In the original PBIL, best-so-far sample (i.e. solution/individual) is used to update PV. But this update scheme may cause serious prematurity very easily, which is because depending on a single sample might lead the search to local optima quickly. A number of improved PV update schemes are thus developed including the famous Hebbian-inspired rule [17]. This rule defines that PV should be updated by a set of samples obtained during the evolution rather than a single sample and to some extent helps maintain a relatively high level of diversity.

The new PV update scheme is based on the Hebbian-inspired rule. Different from other variants, the proposed scheme operates an external population (EP) which only records the best-so-far samples obtained during the evolution. At each generation, a subset of samples is randomly selected from EP and their statistical information is then extracted and used to update PV.

Let $\boldsymbol{P}(t)$ and N denote the PV at generation t and the number of samples generated at each generation, respectively. Let $\boldsymbol{SS}_{\mathrm{EP}} = \{\boldsymbol{B}_1, \ldots, \boldsymbol{B}_M\}$ be the subset of samples randomly selected from EP, where M is smaller than N and $\boldsymbol{B}_i$ is the i-th selected sample, $i = 1, \ldots, M$. Denote the probability distribution of all samples in $\boldsymbol{SS}_{\mathrm{EP}}$ and the learning rate by $\boldsymbol{P}_{SS}$ and α, respectively. The update of $\boldsymbol{P}(t)$ is defined in Eq. (8) and Eq. (9).

$$\boldsymbol{P}(t) = (1.0 - \alpha) \cdot \boldsymbol{P}(t-1) + \alpha \cdot \boldsymbol{P}_{SS} \tag{8}$$

$$\boldsymbol{P}_{SS} = \frac{1}{M} \sum_{k=1}^{M} \boldsymbol{B}_k \tag{9}$$

By introducing extra uncertainty to the PV update process, the proposed scheme helps guide the search exploring unknown areas in the search space as much as possible where optimum may reside. With a certain level of diversity preserved, it helps to prevent the search getting stuck at local optima and hence improves global exploration.

3.2 The Overall Procedure of the Proposed PBIL

In the literature, the binary link state (BLS) individual representation has been widely adopted when tackling network coding related optimization problems [9,10,15,16,18,19]. The proposed PBIL is also based on BLS encoding. Details can be found in [9,10].

Assume there is an individual (sample) X. In terms of fitness evaluation, we first check the feasibility of X. We call X feasible if it results into a valid NCM subgraph $G_{s\rightarrow T}(X)$ and infeasible otherwise. If X is feasible, its fitness value is set to the average BUR in G; otherwise, its fitness value is set to a sufficiently large number (100% in this paper since BUR is in the range [0, 100%]). Note that a feasible individual requires more time than infeasible ones for evaluation since those feasible not only undergo feasibility checking but also NCM subgraph construction. And constructing a NCM subgraph consumes more time than feasibility checking.

```
Initialization
    Set $t$ = 0
    Set $\boldsymbol{P}(t)$ = {0.7,...,0.7}
    Generate $N$ samples by sampling $\boldsymbol{P}(t)$ and put them in set $\mathbf{H}(t)$
    Put the $N$ samples into EP
repeat
    Set $t$ = $t$ + 1
    Evaluate all samples in $\mathbf{H}(t-1)$
    Update EP by finding $N$ best samples from $\mathbf{H}(t-1)$ and EP
    Randomly select $M$ samples from EP and put them into $\boldsymbol{SS}_{\mathrm{EP}}$
    Update $\boldsymbol{P}(t)$ by using the proposed PV update scheme in Subsection 3.1
    Mutate $\boldsymbol{P}(t)$ by Eq.(10)
    Generate $N$ samples by $\boldsymbol{P}(t)$
until termination condition is met
```

Fig. 1. Overall procedure of the proposed PBIL

The overall procedure of the proposed PBIL is shown in Fig. 1. PV $\boldsymbol{P}(t)$ is initialized as {0.7,…,0.7}, meaning the probability of generating ‘1’ at each position in $\boldsymbol{P}(t)$ is 0.7. This helps guide the search to explore promising areas in the search space. Set $\mathbf{H}(t)$ is the sampling set generated by PV at each generation. Note that an all-one individual is inserted into $\mathbf{H}(t)$ to guarantee that the search begins with a feasible sample [10]. The initial N samples are copied into **EP** to form the external population. In the main loop, each sample in $\mathbf{H}(t\text{-}1)$ is evaluated and associated with a fitness value. Then, **EP** is updated. Later on, M random samples are picked up from **EP** and used to update $\boldsymbol{P}(t)$ ($M = N/2$ in this paper).

Mutation operation is implemented to add probability disturbance to PV so that prematurity is alleviated. Let the mutation probability and the amount of mutation at each position in PV, P_k, denoted by p_m and σ, respectively. Let RND_k be a random number (either 0.1 or 1.0, both with a probability of 0.5). If $RND_k < p_m$, P_k is mutated by Eq. (10)

$$P_k = (1.0 - \sigma) \cdot P_k + RND_k \cdot \sigma \quad (10)$$

The termination criterion is that the evolution reaches a predefined number of generations.

4 Performance Evaluation

This section evaluates the proposed PBIL by comparing it with a number of state-of-the-art EAs with respect to the solution quality.

4.1 Test Instances

First of all, test instances are introduced briefly. We consider 12 benchmark instances including 2 fixed and 10 randomly generated directed networks. These instances have been widely adopted for performance evaluation when handling with network coding related optimization problems [8-10,15,16,18-20]. The two fixed networks are F1 (i.e. 7-copy) and F2 (i.e. 15-copy) networks, where each copy is a modified butterfly network and *x*-copy network is constructed by cascading a number of identical copies [18]. As for the random instances, the network scale is from 20 to 60 nodes.

Table 1 shows all instances for performance comparison. In all experiments, each link is with an identical maximum bandwidth, i.e. 100Mbps. The consumed bandwidth of link $e_i \in E$ prior to NCM is randomly generated in the range [1, 50] Mbps with a uniform distribution. The bandwidth consumption of each link in NCM subgraph is set to 30 Mbps. As mentioned in Section 2, the single objective of the problem concerned in the paper is to minimize the average BUR. A smaller average BUR represents a better network load balancing performance. All experiments were run on a Windows XP computer with Intel(R) Core(TM) E8400 3.0GHz, 2G RAM. For performance comparison, each algorithm is run 20 times on each instance.

4.2 Overall Performance Evaluation

In this subsection, performance comparison is carried out among a number of state-of-the-art EAs on 12 benchmark instances. The following lists all algorithms for comparison.

- **GA1**: GA with BLS representation [9].
- **GA2**: GA with block transmission state (BTS) representation [10].
- **UMDA**: univariate marginal distribution algorithm (UMDA) in [21]. Different from PBIL, UMDA utilizes statistics of the last generation to generate a new generation of samples.

Table 1. Benchmark instances for performance comparison

Instances	Nodes	Links	Receivers	Rate
F1	57	84	8	2
F2	121	180	16	2
R1	20	37	5	3
R2	20	39	5	3
R3	30	60	6	3
R4	30	69	6	3
R5	40	78	9	3
R6	40	85	9	4
R7	50	101	8	3
R8	50	118	10	4
R9	60	150	11	5
R10	60	156	10	4

—**QEA1**: Quantum-inspired evolutionary algorithm (QEA) [22]. Based on BLS representation, QEA1 adopts rotation angle step (RAS) and quantum mutation probability (QMP) to update individuals, where RAS is randomly generated and QMP is based on the current fitness value of the associated individual.

— **QEA2**: Another variant of QEA [23]. Different from QEA1, QEA2 modifies the values of RAS and QMP according to the current and previous fitness values of the associated individual.

— **PBIL1**: PBIL devised for the network coding resource minimization problem [16].

— **PBIL2**: The proposed PBIL with the novel PV update scheme in Section 3.

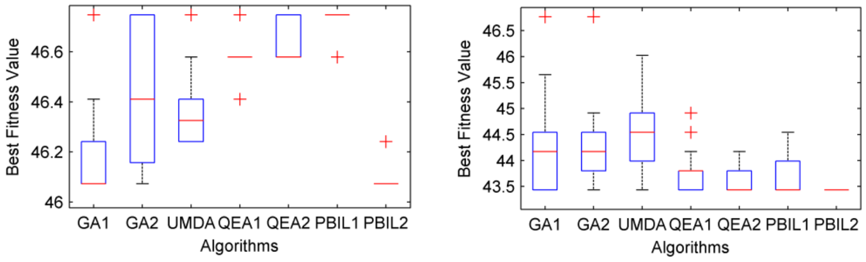
All EAs above uses BLS representation except GA2. The population size and the number of iterations are set to 20 and 200 for each algorithm, respectively. We adopt suggested parameter settings for GA1, GA2, UMDA, QEA1, QEA2, and PBIL1 [9,10,21-23,16]. In PBIL2, we set the learning rate α = 0.1, the mutation probability p_m = 0.02, and the probability variance at each position σ = 0.05, respectively. The number of random samples selected at each generation from EP is 10. All results are collected by running each algorithm 20 times.

Table 2 shows the results of mean value (%) and standard deviation (SD). It is clearly seen that the proposed algorithm, PBIL2, performs the best if considering all test instances. For each instance, PBIL2 obtains the smallest mean value and promising SD, indicating that PBIL2 has a stabilized and outstanding optimization performance in finding near-optimal solutions. By building an evolving probabilistic model, PBIL2 can generate promising samples from the PV at a relatively high probability, which to a certain extent guides the search towards promising areas in the search space. The novel PV update scheme helps PBIL2 gain better global exploration ability and avoid prematurity since the PV update is no longer dependent on a single sample but a random set of samples at each generation. With global exploration enhanced, PBIL2 has more opportunity to reach the global optima and thus obtains the best performance. Fig. 2 shows the box plots of the seven algorithms in six selected instances including F2, R1, R3, R5, R7, and R9. One can find that PBIL2 always outperforms the rest of the EAs for comparison.

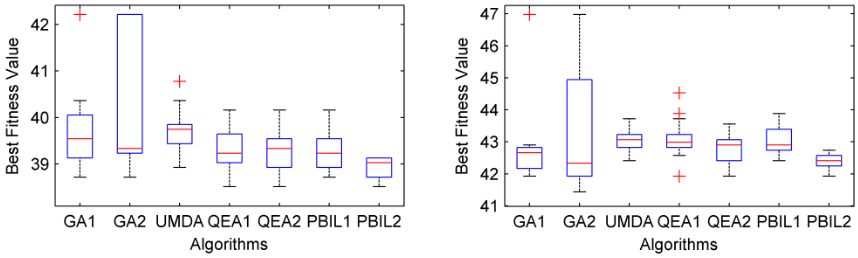
Table 2. Results of mean value (%) and SD (Best results are in bold)

Instances	GA1	GA2	UMDA	QEA1	QEA2	PBIL1	PBIL2
F1	45.33	45.29	**45.11**	45.17	45.35	45.31	**45.11**
	(0.34)	(0.32)	(0.00)	(0.13)	(0.29)	(0.22)	(0.00)
F2	46.20	46.41	46.37	46.57	46.74	46.74	**46.10**
	(0.24)	(0.28)	(0.15)	(0.09)	(0.08)	(0.04)	(0.06)
R1	44.25	44.49	44.54	43.78	43.64	43.69	**43.43**
	(0.86)	(1.06)	(0.68)	(0.41)	(0.25)	(0.40)	(0.00)
R2	41.73	42.24	41.58	41.36	41.19	40.87	**40.52**
	(1.06)	(1.25)	(0.52)	(0.37)	(0.33)	(0.43)	(0.00)
R3	39.95	40.32	39.73	39.33	39.25	39.26	**38.89**
	(1.22)	(1.48)	(0.46)	(0.47)	(0.42)	(0.41)	(0.26)
R4	34.88	34.56	34.82	34.45	34.54	34.46	**34.42**
	(0.62)	(0.40)	(0.37)	(0.23)	(0.25)	(0.25)	(0.19)
R5	43.12	43.38	43.03	43.09	42.80	43.09	**42.40**
	(1.70)	(2.16)	(0.38)	(0.53)	(0.43)	(0.42)	(0.22)
R6	42.69	42.83	42.97	42.62	42.66	42.47	**42.40**
	(0.49)	(0.63)	(0.41)	(0.27)	(0.31)	(0.15)	(0.05)
R7	38.73	38.43	38.09	38.14	38.18	38.26	**37.09**
	(1.72)	(1.79)	(0.52)	(0.62)	(0.45)	(0.39)	(0.41)
R8	40.89	41.59	40.76	41.56	41.51	41.39	**40.28**
	(0.93)	(1.64)	(0.44)	(0.41)	(0.47)	(0.56)	(0.24)
R9	39.38	39.62	39.48	39.79	39.85	39.92	**39.26**
	(0.17)	(0.36)	(0.22)	(0.20)	(0.20)	(0.19)	(0.09)
R10	36.32	36.45	36.20	36.50	36.36	36.66	**36.05**
	(0.89)	(0.95)	(0.29)	(0.31)	(0.37)	(0.30)	(0.14)

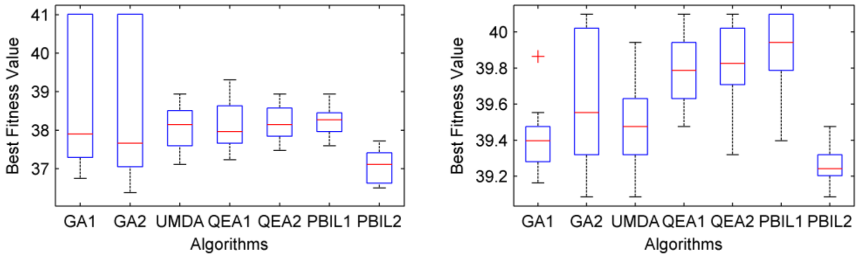
The average computational time (ACT) is another important performance indicator when evaluating EAs. Table 3 shows ACT values of all algorithms in all instances. On the one hand, it can be observed that PBIL2 obtains a relatively large ACT, which means it takes long time for navigating each evolutionary search. This is because PBIL2 does well in global exploration and feasible samples are generated at a high probability. As mentioned in Subsection 3.2, feasible samples consume more time than infeasible ones. If more feasible samples are generated during the evolution, an algorithm definitely incurs more computational cost and hence more ACT. With the novel PV update scheme integrated, PBIL2 generates significant number of feasible samples and hence consumes relatively large amount of ACT. On the other hand, PBIL2 is not the one with the largest ACT either, compared with the others. One may dramatically reduce ACT of PBIL2 by using parallel computation techniques.



(a) F2 (b) R1



(c) R3 (d) R5



(e) R7 (f) R9

Fig. 2. Box plot of 6 selected instances

5 Conclusions

This paper formulates a load balancing optimization problem in network coding based multicast (NCM), where the objective is to keep the network traffic load as balanced as possible when supporting data transmission of NCM. Average bandwidth utilization ratio is used to measure to what extent network traffic load is balanced. To handle with the problem above, we present a modified PBIL with a novel probability vector (PV) update scheme. This scheme introduces additional flexibility in guiding the search by updating the PV by a number of random best samples, which helps enhance the global exploration capability and prevent the search from local optima. Experimental results show that the proposed PBIL gains the best optimization performance compared with a number of state-of-the-art evolutionary algorithms regarding the quality of the solution obtained. Due to the advantages of the proposed algorithm, it can be applied to a number of optimization problems in wireless communications, e.g. traffic control and network configuration in Long Term Evolution (LTE) cellular systems [24,25].

Table 3. Results of ACTs (sec.)

Instances	GA1	GA2	UMDA	QEA1	QEA2	PBIL1	PBIL2
F1	2.2	2.6	2.9	2.2	1.7	0.7	2.6
F2	15.4	12.3	18.1	6.1	4.4	3.1	15.2
R1	0.9	1.0	1.0	1.3	1.0	0.6	1.0
R2	0.9	0.8	1.1	1.3	1.0	0.5	1.1
R3	2.1	1.9	2.5	2.6	2.2	1.1	2.5
R4	2.5	2.8	2.7	3.4	2.6	1.5	2.7
R5	3.6	3.4	3.9	3.0	2.8	1.1	3.4
R6	2.9	2.5	2.9	2.1	1.5	1.2	2.6
R7	4.8	4.9	5.9	5.2	4.6	2.3	5.7
R8	8.8	5.8	8.7	5.3	4.7	2.2	7.2
R9	14.9	10.4	14.5	7.6	7.3	4.4	13.0
R10	13.4	12.6	15.4	14.3	12.8	7.5	15.1

Acknowledgements

This research was supported in part by NSFC (No.61401374), the Fundamental Research Funds for the Central Universities (No. 2682014RC23), the Project-sponsored by SRF for ROCS, SEM, P. R. China and University of Nottingham, UK.

References

1. Benslimane A.: Multimedia multicast on the internet. ISTE (2007).
2. Li, S.Y.R., Yeung, R.W., Cai, N.: Linear network coding. IEEE Trans. Inform. Theory 49(2) (2003) 371-381
3. Wang N., Pavlou G.: Traffic engineered multicast content delivery without MPLS overlay. IEEE Trans. Multimedia, 9 (3) (2007) 619-628.
4. Chi K., Yang C., Wang X.: Performance of network coding based multicast. IEE Proc.-Commun. 153 (3) (2006) 399-404.
5. Hou I.H., Tsai Y.E., Abdelzaher T.F., Gupta I.: AdapCode: adaptive network coding for code updates in wireless sensor networks. In: Proc. INFOCOM. (2008)
6. Vieira F., Lucani D.E., Alagha N.: Codes and balances: multibeam satellite load balancing with coded packets. In Proc. ICC. (2012)
7. Jiang D., Xu Z., Li W., Chen Z.: Network coding-based energy-efficient multicast routing algorithm for multi-hop wireless networks. The Journal of Systems and Software 104 (2015) 152-165.
8. Kim, M., Ahn, C.W., Médard, M., Effros, M.: On minimizing network coding resources: an evolutionary approach. In: Proc. NetCod. (2006)

9. Kim, M., Médard, M., Aggarwal, V., O'Reilly, V., Kim, W., Ahn, C.W., Effros, M.: Evolutionary approaches to minimizing network coding resources. In: Proc. INFOCOM. (2007)
10. Kim, M., Aggarwal, V., O'Reilly, V., Médard, M., Kim, W.: Genetic representations for evolutionary minimization of network coding resources. In: Proc. EvoCOMNET. (2007)
11. Folly K.A.: Multimachine power system stabilizer design based on a simplified version of genetic algorithms combined with learning. In Proc.ISAP2005. (2005)
12. Yang S., Yao X.: Population-based incremental learning with associative memory for dynamic environments. IEEE Trans. Evolut. Comput. 12 (5) (2008) 542-561.
13. Kim J. H., Kim Y. H., Choi S. H., Park I. W.: Evolutionary multi-objective optimization in robot soccer system for education. IEEE Comput. Intell. Magazine 4 (1) (2009) 31-41.
14. Ho S.L., Yang S., Bai Y., Huang J.: A robust metaheuristic combing clonal colony optimization and population-based incremental learning methods. IEEE Trans. Magnetics 50 (2) (2014). DOI: 10.1109/TMAG.2013.2283886.
15. Xing H., Qu R.: A population based incremental learning for network coding resources minimization. IEEE Commun. Letters 15 (7) (2011) 698-700.
16. Xing H., Qu R.: A population based incremental learning for delay constrained network coding resource minimization. In Proc. EvoApplications. (2011)
17. Gonzalez C., Lozano J.A., Larranaga P.: Analyzing the population based incremental learning algorithm by means of discrete dynamical systems. Complex Systems 12 (2000) 465-479.
18. Ahn C. W., Yoo J. C.: Multi-objective evolutionary approach to coding-link cost trade-offs in network coding. Electronics Letters 48 (25) (2012) 1595-1596.
19. Xing H., Qu R.: A nondominated sorting genetic algorithm for bi-objective network coding based multicast routing problems. Inform. Sciences 233 (2013) 36-53.
20. Xing H., Qu R., Bai L., Ji Y.: On minimizing coding operations in network coding based multicast: an evolutionary algorithm. Applied Intelligence 41(3) (2014) 820-836.
21. Lozada-Chang L.V., Santana R.: Univariate marginal distribution algorithm dynamics for a class of parametric functions with unitation constraints. Inform. Sciences 181 (11) (2011) 2340-2355.
22. Xing H., Ji Y., Bai L., Sun Y.: An improved quantum-inspired evolutionary algorithm for coding resource optimization based network coding multicast scheme. AEUE 64 (12) (2010) 1105-1113.
23. Ji Y., Xing H.: A memory-storable quantum-inspired evolutionary algorithm for network coding resource minimization. In Kita E. (Ed.), Evolutionary Algorithm, InTech 363-380. (2011)
24. Xu L., Chen Y., Chai K. K., Schormans J., Cuthbert L.: Self-organising cluster-based cooperative load balancing in OFDMA cellular networks. Wiley Wireless Commun. Mobile Comput. 15 (7) (2015), 1171-1187.
25. Xu L., Cheng X., Chen Y., Chao K., Liu D., Xing H.: Self-optimised coordinated traffic shifting scheme for LTE cellular systems. In Proc. ICSON2015. (2015)