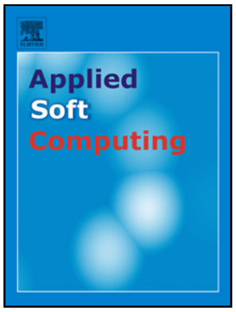
Accepted Manuscript



Title: An Improved MOEA/D Algorithm for Multi-objective Multicast Routing with Network Coding

Authors: Huanlai Xing, Zhaoyuan Wang, Tianrui Li, Hui Li, Rong Qu

PII:	S1568-4946(17)30296-X
DOI:	http://dx.doi.org/doi:10.1016/j.asoc.2017.05.033
Reference:	ASOC 4237

To appear in: *Applied Soft Computing*

Received date:	3-7-2015
Revised date:	12-4-2017
Accepted date:	17-5-2017

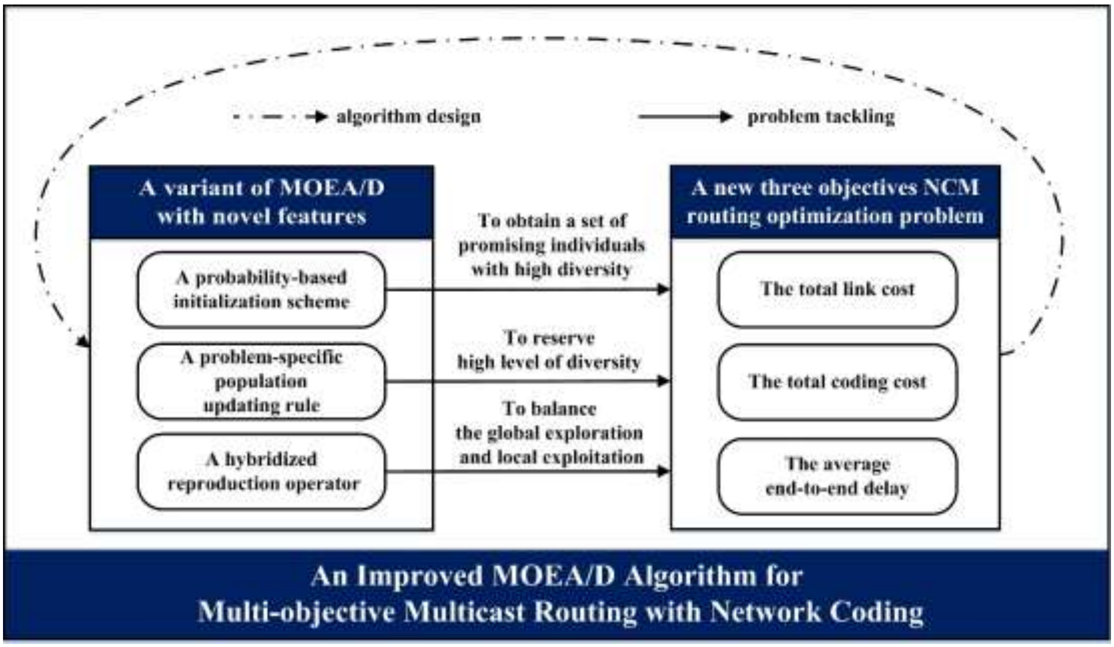
Please cite this article as: Huanlai Xing, Zhaoyuan Wang, Tianrui Li, Hui Li, Rong Qu, An Improved MOEA/D Algorithm for Multi-objective Multicast Routing with Network Coding, Applied Soft Computing Journalhttp://dx.doi.org/10.1016/j.asoc.2017.05.033

An Improved MOEA/D Algorithm for Multi-objective Multicast Routing with Network Coding

Huanlai Xing[1], Zhaoyuan Wang[1], Tianrui Li[1], Hui Li[2], Rong Qu[3]

1. School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, P.R. China
2. School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, P.R. China
3. School of Computer Science, The University of Nottingham, Nottingham NG8 1BB, United Kingdom

Graphical abstract


algorithm design
problem tackling
A variant of MOEA/D with novel features
A probability-based initialization scheme
A problem-specific population updating rule
A hybridized reproduction operator
To obtain a set of promising individuals with high diversity
To reserve high level of diversity
To balance the global exploration and local exploitation
A new three objectives NCM routing optimization problem
The total link cost
The total coding cost
The average end-to-end delay
An Improved MOEA/D Algorithm for Multi-objective Multicast Routing with Network Coding

Highlights

- Formulate a network-coding-based multicast routing problem with three objectives.
- Propose a hybrid multi-objective evolutionary algorithm incorporating PBIL into MOEA/D.
- Develop a probability-based initialization scheme, a problem-specific population updating rule, and a hybridized reproduction operator to enhance the optimization performance.

Abstract

Network coding enables higher network throughput, more balanced traffic, securer data transmission, etc. However, complicated mathematical operations incurs when recombining packets at intermediate nodes, which if not operated properly, lead to very high network resource consumption and unacceptable delay. Therefore, it is of vital importance to minimize various network resources and end-to-end delays while exploiting promising benefits of network coding.

Since multicast has been used in increasingly more applications, such as video conferencing and remote education, we study the multicast routing problem with network coding. The problem is formulated as a multi-objective optimization problem (MOP), where the total coding cost, the total link cost and the end-to-end delay are the three objectives for minimization simultaneously. We adapt multi-objective evolutionary algorithm based on decomposition (MOEA/D) for this MOP by hybridizing it with population-based incremental learning technique, which makes use of the global and historical information collected to provide additional guidance to the evolutionary search. Three new schemes are devised to facilitate the performance improvement, including a probability-based initialization scheme, a problem-specific population updating rule, and a hybridized reproduction operator. Experimental results clearly demonstrate that the proposed algorithm outperforms a number of state-of-the-art MOEAs regarding the solution quality and computational time.

Keywords: Network coding; Multicast; Multi-objective evolutionary Algorithm

1. Introduction

Multicast is a one-to-many data delivery method in telecommunications, where information sent from the source is copied and routed to a number of destinations simultaneously. Compared with multiple unicasts, multicast is of higher bandwidth efficiency, especially when there are a large number of receivers [1]. The Internet has witnessed a significant growth in multimedia applications (e.g. video conferencing, IPTV, and remote education), where multicast is a key supporting technique [2]. However, traditional multicast schemes adopt the store-and-forward data forwarding, where the throughput may not reach to the theoretical maximum [3].

Network coding is a newly emerged communications paradigm, where instead of simply copying and forwarding the incoming data, any intermediate node in the network is allowed to perform mathematical operations (e.g. operations over some finite field) to recombine different incoming data if necessary [3]. This technique has been reported to be effective and useful in traffic balancing, data security, energy saving, network tomography, and robustness against failures [4, 5, 6, 7]. In particular, with network coding, multicast can always achieve the theoretical maximum multicast data rate [4].

Fig.1 shows a multicast scenario with respect to the multicast data rate, where traditional routing and network coding are employed separately. Fig.1(a) is the topology of the scenario, where each link is directional and with a capacity of one bit per time unit. Source *s* wants to multicast two bits, *a* and *b*, to two receivers, t_1 and t_2. According to the Max-Flow Min-Cut theorem, the minimum cut between *s* and t_1 (or t_2) is two bits per time unit, so is the maximum data rate from *s* to t_1 and from *s* to t_2. Nevertheless, if traditional routing is adopted, as can be seen in Fig.1(b), bottleneck link K→V would only allow a single bit to be delivered, causing reduction in the data rate. This is because traditional routing is based on the store-and-forward forwarding. On the contrary, if node K can perform mathematical operation to recombine *a* and *b* into a single bit, $a \oplus b$, the theoretical maximum data rate to each receiver can be obtained at the same time, where in the example of Fig.1(c) symbol $\oplus$ is exclusive-OR operation. Nodes t_1 and t_2 can receive $\{a, a \oplus b\}$ and $\{b, a \oplus b\}$ and recover the original information *a* and *b* after calculating $a \oplus (a \oplus b)$ and $b \oplus (a \oplus b)$, respectively. So, the maximum multicast data rate is equal to the theoretical data rate.

1.1 Related work

As aforementioned, network coding brings benefits to multicast. However, in network coding based multicast (NCM), data recombination has to be executed at the network layer by performing complicated mathematical operations (called coding operations) to combine different incoming information at corresponding intermediate nodes. Hence, the computational overhead could be extremely high and complex coding operations may cause large end-to-end delays. In addition, instead of maintaining a single path, NCM employs multiple paths to deliver data to any receiver, which enables super-fast data rate, however, at the expense of higher link resource consumption. Therefore, some research has been conducted to optimize NCM routing from different aspects while utilizing the benefits of network coding. Main research streams on the NCM routing optimization include coding cost minimization, link cost minimization, delay related optimization and multi-objective optimization.

- **Coding cost minimization**. Performing coding operations consumes extra computing resources. Hence, one stream focuses on minimizing the amount of coding operations necessarily performed. Early studies include greedy-based optimization approaches [8, 9] and genetic algorithms (GAs) [10]. Recently, researchers adapt estimation of distribution algorithms (EDAs) for minimizing coding cost, e.g. quantum-inspired evolutionary algorithms [11], population based incremental learning (PBIL) [12] and compact GA [13]. Moreover, Evolutionary algorithms (EAs) with other techniques incorporated, such as entropy-based evaluation relaxation, path-oriented encoding and artificial ant colonies, have been investigated [14, 15, 16].

- **Link cost minimization**. NCM data are delivered through multiple paths [8]. In real networks, different links, when employed for data transmission, incur different costs, known as link costs. NCM routing plan with the least link costs is preferred. Lun et al. formulate a minimum-cost NCM multicast over packet networks, where wireline and wireless networks are both considered [17]. Cui and Ho study the least-cost integral network coding problem, where the packet injection rate on each link is constrained to be integral [18]. Researchers also investigate the minimum cost subgraph construction in static and dynamic environments [19].
- **Delay related optimization**. Delay is one of the most important metrics reflecting network performance. A considerable amount of applications requires guarantee on stringent delay, especially for real-time broadband multimedia applications [20]. However, network coding gains high bandwidth utilization at the expense of extra computational resource consumption in corresponding intermediate nodes [4]. Packet recombination (i.e. coding operation) incurs additional processing delay in individual nodes and causes severely large end-to-end delays if data are not routed appropriately. Delay-related issues have thus drawn a great amount of attention. Delay analysis and its minimization have been studied in the context of wireless networks [21], overlay networks [22], broadcast erasure channels with feedback [23], instantly decodable network coding [24, 25], and multicasting [26].
- **Multi-objective optimization**. All the above researches concern single-objective optimization problems. However, both coding and link costs incur in NCM data delivery and they should be minimized. This problem can be formulated as a bi-objective optimization problem, for which a number of multi-objective evolutionary algorithms (MOEAs) have been proposed [27, 28]. When launching NCM, network service providers (NSPs) pay for computing and bandwidth resources they consume. Optimization on the two objectives helps NSPs to find a trade-off between the computing resource and the bandwidth resource, to gain high profits. In addition, end users usually expect to have decent quality of experience (QoE), especially small end-to-end delay. This conflicts with interests of NSPs expecting less network resource consumption. In [29], the trade-off between the weighted sum of the coding and link costs and the maximum end-to-end delay of multiple paths is studied.

Towards practical deployment of NCM, it is important to study the trade-off between coding and link costs, as well as satisfying end users with high QoE (especially delay). However, such issue has received little consideration. Many existing problem models do not take user experience into account [27]. Models only concerning the minimization of the total cost and end-to-end delay cannot distinguish the trade-off between the coding and link costs [29]. This paper extends the problem models in [27,29], and establishes a new multi-objective NCM routing model, where all the key factors in NCM data transmission, namely the coding cost, the link cost and the average end-to-end delay, are formulated as three objectives for minimization.

1.2 Variants of MOEA/D

Multiobjective optimization evolutionary algorithms (MOEAs) can easily obtain a set of promising solutions in a single run due to their population-based frameworks. They have thus received increasingly more research attention from fields of multi-objective optimization and evolutionary computation. Multi-objective evolutionary algorithm based on decomposition (MOEA/D) is among the highlighted MOEAs [30]. MOEA/D decomposes a multi-objective optimization problem (MOP) into a number of scalar optimization subproblems, each with an aggregated objective. MOEA/D showed to have a better optimization performance with lower computational cost than a number of state-of-the-art MOEAs, e.g. NSGA-II [31] and SPEA2 [32].

In the literature, a number of sophisticated techniques have been incorporated into the MOEA/D framework to further exploit its potential, e.g. estimation of distribution algorithm, differential evolution, memetic algorithm, ant colony optimization, particle swarm optimization, and simulated annealing.

- **Estimation of distribution algorithm (EDA)**. Recently, EDAs gain increasing attentions in solving scalar optimization problems (SOPs) [33]. In principle, they are a family of EAs that incorporate machine learning techniques, where statistical information of promising solutions is extracted to build probabilistic models, from which samples are generated. Compared with traditional EAs, EDAs usually obtain better optimization results, with relatively less computational complexity. Shim et al. integrate the restricted Boltzmann machine and the evolutionary gradient search into MOEA/D [34]. This algorithm performs well when addressing the multiobjective multiple traveling salesman problem (TSP). To gain a balanced performance on global exploration and local exploitation, a hybrid adaptive MOEA that synthesizes GA, EDA and differential evolution is presented [35]. Promising solutions generated at early generation are used to produce a proportion of solutions for the next generation. Giagkiozis et al. hybridize MOEA/D and EDA for many-objective optimization problems [36], where a generalized decomposition method unifies different performance objectives.
- **Differential evolution (DE)**. DE has been integrated into MOEA/D to effectively handle complicated Pareto fronts for MOPs, and it was reported to perform much better than NSGA-II [37]. Tan et al. propose a modified MOEA/D-DE that generates uniformly distributed scalar optimization subproblems and a simplified quadratic approximation for enhancing the local exploitation and the accuracy of aggregation function values [38]. Combined with Gaussian mutation operators, MOEA/D-DE has a decent performance in devising Yagi-Uda antennas [39]. An adaptive DE for multiobjective problem (ADEMO/D), which is featured with a number of adaptive strategies, gains evenly distributed solutions well approximating the Pareto front for continuous MOPs [40].
- **Memetic algorithm (MA)**. Local search operators (maybe assisted with domain knowledge) have recently been incorporated into MOEA/D. With improved local exploitation, the proposed algorithms provide better solutions than pure MOEA/D.

Mei et al. propose a MA based on decomposition with extended neighborhood search, namely D-MAENS, for solving the capacitated arc routing problem [41]. Shang et al. improve the performance of D-MAENS by two novel schemes, one for solution replacement and the other for elitism maintenance [42]. MOEA/D is hybridized with a mathematical programming technique, where Nelder and Mead's algorithm serves as the local search mechanism [43]. Mashwani and Salhi present a hybrid MOEA/D, where particle swarm optimization (PSO) and DE are integrated. DE acts as the main evolutionary framework while PSO is in charge of local search [44]. By combining ideas from MOEA/D and Pareto local search, Ke et al. propose a memetic algorithm based on decomposition (MOMAD) [45], where three populations are initialized by a problem-specific single objective heuristic and evolved by the Pareto local search and single objective local search procedures.

- **Ant colony optimization (ACO)**. Inspired by ACO, Li et al. incorporate a probabilistic representation based on pheromone trails into MOEA/D framework and demonstrate its good potential in handling hard MOPs with many local optima [46]. Ke et al. introduce ACO into MOEA/D for solving multi-objective 0-1 knapsack problem and bi-objective TSP [47]. Instead of using sub-colonies, each ant solves one of the SOPs. Cheng et al. propose MoACO/D, where an ant colony is divided into many sub-colonies overlapping each other, and each sub-colony tackles a certain SOP decomposed from the original MOP [48].
- Other optimization techniques. A considerable amount of research efforts has also been dedicated to integrating other techniques into the MOEA/D framework, such as PSO [49], simulated annealing [50], fuzzy system [51], Gaussian process model [52], opposition-learning [53], teaching-learning algorithm [54], and so on.

With different techniques integrated, hybrid MOEA/Ds are usually reported to gain decent optimization performance when solving MOPs. As one of the EDAs, population based incremental learning (PBIL) combines GA and machine learning. It manipulates a real-valued probability vector (PV) and extracts statistical information from promising samples to evolve the PV [55]. Unlike other EAs, PBIL involves neither explicit population nor complicated operators, such as crossover. Hence, it incurs less computational and memory overhead while gaining similar or even better performance, compared with traditional EAs. Besides, PBIL is an excellent optimizer for solving the NCM-based single-objective optimization problem [12]. We thus explore the potential of integrating PBIL components into MOEA/D when addressing the three-objective NCM routing problem in this paper.

1.3 Contribution of our work

The contribution of the work includes the formulation of a new multi-objective optimization problem and a hybrid MOEA to address it, as listed below.

- **A NCM routing optimization problem with three objectives**. The computing resource, bandwidth resource and delay are all important factors when considering the practical deployment of NCM. All of them need to be kept as low as possible. We formulate a three-objective optimization problem, which minimizes the total coding cost, the total link cost and the average end-to-end delay of NCM.

To tackle the MOP above, we propose a hybrid MOEA incorporating PBIL into the original MOEA/D framework, with three novel features listed as below.

- **A probability-based initialization scheme**. The problem concerned in the paper is highly constrained and infeasible solutions dominate the search space. To start with feasible individuals, we propose a probability-based initialization scheme, where each individual is created according to the estimated distribution of feasible solutions. It helps obtain a set of promising individuals with high diversity.
- **A problem-specific population updating rule**. Due to the special features of the proposed problem, the original MOEA/D may reproduce similar individuals in the population, leading to serious prematurity and thus a deteriorated performance. To overcome this problem, the paper introduces a problem-specific population updating rule. Once a promising individual is generated, it updates a single individual in the population, if the improvement of the individual quality is the most significant among the current population. This helps reserve high level of diversity.
- **A hybridized reproduction operator**. Global exploration and local exploitation are two important research issues when designing effective MOEAs. However, they usually contradict with each other. To strike a balance between them, we devise a reproduction operator which combines reproduction techniques in GA and PBIL. A control function is devised to decide the percentage of individuals generated from each reproduction technique. Analysis indicates that with this operator, the evolution can maintain a relatively high level of global exploration and it thus contributes to a balanced optimization performance.

The rest of the paper is organized below. Section 2 describes the problem formulation. Section 3 briefly reviews the original MOEA/D and PBIL. The proposed algorithm is introduced in Section 4. Simulation results are analyzed in Section 5. Conclusions and future work are in Section 6.

2. Problem formulation

A network is represented by a directed graph $G = (V, E)$, where V is the node set and E is the link set. Each link $e \in E$ in the network is associated with a currently consumed bandwidth $\omega_c(e)$ and a maximum bandwidth $\omega_m(e)$, where $\omega_m(e) \geq \omega_c(e)$. Besides, each link e is associated with a non-negative value $c_{link}(e)$, which is the cost per unit flow on this link. For simplicity, we assume only one unit flow, e.g. 10Mb/s, is allowed on each link for transmitting NCM data. That is, if a link is employed by a NCM session, bandwidth of one unit flow, ω_{uf}, is consumed.

In real-world networks, value $c_{link}(e)$ could be the cost for leasing link e, where network operators and service providers should determine the cost for each link according to the capital expenditure (CAPEX) and operating expenditure (OPEX); or, $c_{link}(e)$ could be in proportion to the bandwidth utilization ratio (BUR) of e so that links with smaller BURs are more likely to be employed, which is in favor of load balancing.

A single-source NCM session can be defined as a 4-tuple set (G, s, T, R), where a source node $s \in V$, would like to have identical data delivered to a set of receivers $T = \{t_1,\ldots, t_d\}$ at an expected data rate R [4, 6]. We refer to link-disjoint paths as paths without any common link. For an arbitrary receiver, if R link-disjoint paths can be constructed, the data rate R is achievable. In this paper, linear network coding is adopted as it is simple yet sufficient for supporting NCM [4].

Given a NCM request, the task is to find a connected subgraph in G to support the multicast with network coding [13]. This subgraph is referred to as a *NCM subgraph* (denoted by $G_{NCM}(s,T)$). A NCM subgraph includes R link-disjoint paths connecting s and each receiver. This subgraph is composed of $R\cdot|T|$ paths, i.e. $\{p_i(s, t_k)| \, i = 1,\ldots,R, k = 1,\ldots, |T|\}$. A *coding node* is a node which performs coding operations; a *coding link* is an outgoing link of a coding node via which the outgoing data are a mathematical combination of the data received by the coding node. In network G, a *merging node* is a non-receiver intermediate node with multiple incoming links [10]. Only merging nodes can become coding nodes and perform packet recombination. The number of coding links is a precise estimator of how many coding operations to be performed [9]. So, we use it to estimate the amount of coding operations in a NCM subgraph. More descriptions can be found in [13].

For an arbitrary path $p_i(s, t_k)$, it is a chain of nodes and links that originates from s and terminates at t_k. Data buffering, processing and forwarding in nodes will incur the processing delay. There are two types of intermediate nodes, namely the forwarding nodes and the coding nodes. Forwarding nodes simply store and forward the incoming flows while coding nodes perform coding operations and send out the coded flows. Coding nodes usually result into significantly longer processing time than the forwarding ones. This is not only because if a number of data packets are to be coded together at a certain coding node, the coding operation cannot start until the arrival of the last data packet (i.e. the buffering time), but also because the complicated mathematical calculations over finite field are performed (i.e. the processing time). So, we simply ignore the processing delay incurred in forwarding nodes. For coding nodes, we consider the buffering time for waiting for the arrival of the last data packet for coding and the processing time for performing a coding operation. Besides, a receiver usually needs to decode the received coded information to recover the original information sent from the source. However, this is trivial compared with the accumulated delay from the source to the receiver. We thus ignore the decoding delay incurred at receivers. On the other hand, each link along $p_i(s, t_k)$ will incur the propagation delay that is proportional to the distance it spans.

The following lists notations used in the paper:

- s: the source node in network $G(V, E)$;
- $T = \{t_1, t_2, \ldots, t_d\}$: the set of receivers, where $d=|T|$ is the number of receivers;
- R: the data rate (an integer) at which s delivers data to T;
- $p_i(s, t_k)$: the i-th path from s to t_k, where $t_k \in T$ and $i=1, \ldots, R$;
- $r(s, t_k)$: the achievable data rate from s to receiver $t_k \in T$;
- σ_e: a binary variable associated with link $e \in G_{NCM}(s, T)$. If e is a coding link, we set $\sigma_e = 1$; otherwise, $\sigma_e = 0$.

- $c_{link}(e)$: the cost incurred on link e if $e \in G_{NCM}(s, T)$;
- $delay_{node}(v)$: the processing delay incurred in node v;
- $delay_{link}(e)$: the propagation delay incurred on link e;
- $delay(p_i(s, t_k))$: the end-to-end delay of path $p_i(s, t_k)$;
- $\Phi(p_i(s, t_k))$: the link set of path $p_i(s, t_k)$.

The task is to find an appropriate NCM subgraph in $G(V, E)$, which satisfies the NCM data rate requirement, with three objectives minimized, as shown below.

$$\text{Minimize: } \begin{cases} f_1 = \sum_{\forall e \in G_{NCM}(s,T)} c_{link}(e) \\ f_2 = \sum_{\forall e \in G_{NCM}(s,T)} \sigma_e \\ f_3 = \dfrac{1}{d} \sum_{k=1}^{d} \max\{delay(p_i(s,t_k)) \mid i = 1, \ldots R\} \end{cases} \tag{1}$$

$$delay(p_i(s,t_k)) = \sum_{\forall v \in p_i(s,t_k)} delay_{node}(v) + \sum_{\forall e \in p_i(s,t_k)} delay_{link}(e),\ \forall t_k \in T \tag{2}$$

$$\text{Subject to: } r(s,t_k) = R,\ \forall t_k \in T \tag{3}$$

$$\omega_m(e) \geq \omega_c(e) + \omega_{uf},\ \forall e \in G_{NCM}(s,T) \tag{4}$$

$$\Phi(p_i(s,t_k)) \cap \Phi(p_j(s,t_k)) = \varnothing,\ \forall i, j \in \{1, \ldots, R\}, i \neq j \tag{5}$$

In Objective (1), objective f_1 is to minimize the bandwidth resource, i.e. the total link cost of the NCM; objective f_2 is to minimize the computing resource consumption, i.e. the total coding cost of the NCM; objective f_3 is to minimize the average end-to-end delay along all paths in the NCM subgraph. Equation (2) defines the end-to-end delay along $p_i(s, t_k)$. Constraint (3) defines that R paths are to be constructed from s to each of the receivers. Constraint (4) is the bandwidth constraint, where for an arbitray link $e \in G_{NCM}(s, T)$, its bandwidth consumption cannot exceed the maximum bandwidth. Constraint (5) explains that for an arbitrary receiver t_k, any two paths cannot share a common edge.

Fig.2 illustrates an example NCM scenario, where source s delivers two bits, a and b, to two receivers, t_1 and t_2, respectively. The data transmission scheme is shown in Fig.2(a), where coding node K performs packet recombination $a \oplus b$. Fig.2(b) shows the four paths originating from source s to one of the receivers. Note that, paths to the same receiver are link-disjoint paths, e.g. $p_1(s \rightarrow t_1)$ and $p_2(s \rightarrow t_1)$. We have the propagation delays over all links as shown in Fig.2(c). Suppose the processing time for performing a coding operation is 1*ms* and there is no delay incurred for simple forwarding. We calculate the end-to-end delays along each path. For paths $p_1(s \rightarrow t_1)$ and $p_1(s \rightarrow t_2)$, as there is no coding node involved, their end-to-end delays are simply the summations of the propagation delays along them, respectively. So, we have $delay(p_1(s \rightarrow t_1))$ = 12*ms* and $delay(p_1(s \rightarrow t_2))$ = 13*ms*. For $p_2(s \rightarrow t_1)$ and $p_2(s \rightarrow t_2)$, node K is the coding node (where a and b are recombined). Data a and b arrive K after 6*ms* and 9*ms*, respectively. The former must wait for the latter for recombination. After that, the coded data is output via link K→V and then arrive at t_1 and t_2, respectively.

Hence, we have *delay*($p_2(s \rightarrow t_1)$) = 17*ms* and *delay*($p_2(s \rightarrow t_2)$) = 15*ms*.

The above three-objective minimization problem is a MOP. Suppose there are two solutions (f_1^*, f_2^*, f_3^*) and ($f_1^{'}$, $f_2^{'}$, $f_3^{'}$). (f_1^*, f_2^*, f_3^*) dominates ($f_1^{'}$, $f_2^{'}$, $f_3^{'}$) or ($f_1^{'}$, $f_2^{'}$, $f_3^{'}$) is dominated by (f_1^*, f_2^*, f_3^*) only if any of the following three conditions is satisfied: {$f_1^* < f_1'$, $f_2^* \leq f_2'$, $f_3^* \leq f_3'$ } or {$f_1^* \leq f_1'$, $f_2^* < f_2'$, $f_3^* \leq f_3'$ } or {$f_1^* \leq f_1'$, $f_2^* \leq f_2'$, $f_3^* < f_3'$ }. Optimal solutions to the problem above constitute the Pareto-optimal Set (PS) in the decision space and their mapping forms the Pareto-optimal Front (PF) in the objective space.

3 Overview of MOEA/D and PBIL

3.1 MOEA/D

The fundamental idea of MOEA/D is to decompose a MOP into N scalar optimization subproblems (SOSPs) [30]. MOEA/D aims at optimizing all SOSPs simultaneously in a collaborative and time-efficient manner. Three decomposition methods are introduced in [30]. This paper considers the Tchebycheff approach, one of the most commonly used. A SOSP achieved by the decomposition of a MOP can be expressed in Eq. (6).

$$Minimize: \; g(x \mid \lambda, z^*) = \max\nolimits_{1 \leq j \leq m} \{\lambda_j \mid f_j(x) - z_j^* \mid\}, \forall \; x \in \Omega \qquad (6)$$

where m is the number of objectives, $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_m)$ is a weight vector, $\lambda_j \geq 0$, j = 1, …, m, and $\sum_{j=1}^{m} \lambda_j = 1$. In reference point $z^*=(z_1^*, \ldots, z_m^*)$, $z_j^* = \min \{f_j(x) \mid x \in \Omega\}$, where Ω is the decision space.

It is assumed that a set of N weight vectors $\boldsymbol{\lambda}^1, \ldots, \boldsymbol{\lambda}^N$ should be selected properly so the optimal solutions of those SOSPs will well approximate the PF. In addition, the neighborhood relationship of SOSPs can be measured by Euclidean distances between the weight vectors. Neighboring SOSPs have similar fitness landscapes and their optimal solutions should be close in the decision space. Information sharing between neighborhoods can be exploited to accomplish the optimization task. The evolutionary procedure of MOEA/D is described below.

Global structure:

— A population of N points $x_1, \ldots, x_N \in \Omega$, where x_i is the individual to SOSP(i), the i-th SOSP.

— A reference point $z = (z_1, \ldots, z_m)$, where z_j , j =1,…, m, is the best-so-far value of objective f_j.

－ An external population (EP), which stores nondominated solutions found during the search.

Input: N: number of SOSPs; W: number of the neighbors for each SOSP; $\boldsymbol{\lambda}^1, \ldots, \boldsymbol{\lambda}^N$: uniformly distributed weight vectors; p_c: crossover rate; p_m: mutation rate.

```
MOEA/D Procedure:
Initialization
1.  Set EP = ∅.
2.  For each $\lambda^i$, calculate the $W$ closest weight vectors, $\lambda^{i(1)}, \ldots, \lambda^{i(W)}$, by Euclidean
    distance and set $\varphi(i) = \{i(1), \ldots, i(W)\}$.
3.  Generate an initial population $x_1, \ldots, x_N$ and evaluate $f_u(x_j)$ for each individual.
4.  Initialize $z = (z_1, \ldots, z_m)$.
Repeat
5.  For $i$ =1 to $N$ do
6.      Reproduction: Generate a new solution $y$ by two individuals $x_u$ and $x_l$ using
        crossover and mutation operators, where $u, l \in \varphi(i)$.
7.      Improvement: Improve $y$ by using a problem-specific improvement repair
        operator, which is optional.
8.      Update of z: For $j = 1, \ldots, m$, if $f_j(y) < z_j$, set $z_j = f_j(y)$.
9.      Update of neighboring solutions: For each $k \in \varphi(i)$, if $g(y|\lambda^k, z) \leq g(x_k|\lambda^k, z)$,
        then set $x_k = y$ and $f_j(x_k) = f_j(y)$, $j = 1, \ldots, m$.
10.     Update of EP: Remove those solutions dominated by $y$ from EP and add $y$ to
        EP if it is not dominated by any member in EP.
Termination
11. Until stopping criteria are satisfied, output EP.
```

3.2 PBIL

PBIL has been reported to gain promising optimization performance when solving the single-objective network coding resource minimization problem [12]. Instead of using an explicit population, PBIL manipulates a real-valued probability vector (PV). When sampled, PV generates a number of binary solutions and the best one is used to update the PV. By making use of global information, promising solutions are generated with increasingly higher probabilities stored in PV.

Let $\mathbf{P}^{(k)} = (P_1^k, \ldots, P_L^k)$ be PV at generation k, where L is the individual length and P_l^k is the probability of obtaining '1' at the l-th position. Denote $\mathbf{B}^{(k)} = (B_1^k, \ldots, B_L^k)$ and α the best so far solution during the search and the learning rate, respectively. Fig.3 shows the pseudo-code of the original PBIL. The PV at generation k, $\mathbf{P}^{(k)}$, is updated by Eq. (7). Mutation operation may be used to avoid local optima [55]. Let σ be the probability shifting at each position, and P_l^k is to be mutated, Eq. (8) is typically adopted in mutation.

$$\mathbf{P}^{(k)} = (1.0 - \alpha) \cdot \mathbf{P}^{(k-1)} + \alpha \cdot \mathbf{B}^{(k)}, \qquad (7)$$

$$P_l^k = (1.0 - \sigma) \cdot P_l^k + f_{rnd} \cdot \sigma \qquad (8)$$

where f_{rnd} is either 0.0 or 1.0, randomly generated with a probability of 0.5.

4 The proposed MOEA/D-PBIL

This section begins from the individual representation and evaluation. Then, three novel schemes, i.e. a probability-based initialization scheme, a problem-specific population updating rule and a hybridized reproduction operator, are introduced. At last, the overall pseudo-code of the proposed MOEA/D-PBIL is given in detail.

4.1 Individual representation and evaluation

As known, the individual representation is one of the most important issues in EAs. Binary link state individual representation (BLS-IR) has been widely used in network coding related optimization problems, including a number of single-objective optimization problems and MOPs [10, 11, 12, 13, 29]. In particular, BLS-IR is able to facilitate an easy process of estimating the consumption of the coding resource in NCM data transmission. As mentioned before, coding operations can be performed at merging nodes only. BLS-IR is based on the graph decomposition method (GDM) which helps to clearly show how information flows are forwarded within each merging node [10]. The MOP in this paper also involves the estimation of coding resource consumption, so it is rationale to use BLS-IR to represent individuals. The following introduces GDM, BLS-IR and the raw fitness calculation.

In GDM, each merging node M with I_M incoming links and O_M outgoing links is decomposed into I_M incoming auxiliary nodes and O_M outgoing auxiliary nodes, connected by all possible routes passing the merging node. Each link flows into node M is redirected to one of the I_M incoming auxiliary nodes, and each node has only one link flows into it. Similarly, each outgoing link from node M is redirected to one of the O_M outgoing auxiliary nodes and each node has only one outgoing link. Besides, within each decomposed merging node, an auxiliary link connects each incoming auxiliary node with each outgoing auxiliary node. Given an original graph *G*, every merging node is decomposed by GDM and then a decomposed graph *G'* is created.

In BLS-IR, each individual *x* is represented by a string of binary variables, each associated with an auxiliary link between auxiliary nodes. Value 1 for a binary variable means the corresponding link in *G'* is active and information can pass by; otherwise, the corresponding link in *G'* is inactive and information transmission is not allowed. An individual *x* thus corresponds to an explicit and unique decomposed graph $G_D(x)$. Based on $G_D(x)$, we determine if a valid NCM subgraph (see Section 2 for details) can be found.

An example of the individual representation is shown in Fig.4. Fig.4(a) is the original graph, where source *s* would like to deliver data to receivers t_1 and t_2 at a data rate of 2 unit flows. There are two merging nodes, K and V, because they are non-receiver intermediate nodes with more than one incoming link. Take K as an example, as it has 2 incoming links and 2 outgoing links, GDM decomposes it into 2 incoming auxiliary nodes (k_1 and k_2) and 2 outgoing auxiliary nodes (k_3 and k_4). An auxiliary link is created from each incoming auxiliary node to each outgoing auxiliary node. So, there are 4 auxiliary links, namely $k_1 \rightarrow k_3$, $k_2 \rightarrow k_3$, $k_1 \rightarrow k_4$, and $k_2 \rightarrow k_4$. Node V is decomposed in the same way. The decomposed graph is shown in Fig.4(b). After decomposition, we have 8 auxiliary links and thus use 8-bit binary string to represent an individual. Suppose the states of the 8 auxiliary links are in the following order: $k_1 \rightarrow k_3$, $k_2 \rightarrow k_3$, $k_1 \rightarrow k_4$, $k_2 \rightarrow k_4$, $v_1 \rightarrow v_3$, $v_2 \rightarrow v_3$, $v_1 \rightarrow v_4$, and $v_2 \rightarrow v_4$. If there is an individual '11001010', the states of the corresponding auxiliary links are illustrated in Fig.4(c). Then, we have four active auxiliary links that allow information to pass, including $k_1 \rightarrow k_3$, $k_2 \rightarrow k_3$, $v_1 \rightarrow v_3$, and $v_1 \rightarrow v_4$, as shown in Fig.4(d). Based on the explicit decomposed graph, we can easily find two link-disjoint paths for each receiver (see Fig.4(e)). Finally, the NCM subtraph is constructed in Fig.4(f).

When evaluating an individual *x*, its feasibility is first checked. If a NCM subgraph with the expected data rate satisfied can be found from $G_D(x)$, *x* is feasible; otherwise, it is regarded infeasible. One of the max-flow algorithms, the Goldberg algorithm, is used to calculate the max flow between the source and each receiver within the obtained NCM subgraph [56]. For feasible individuals, the objective values are calculated according to Eq. (1). For infeasible ones, three sufficiently large values are set to the objective values, ensuring that the infeasible is less competitive than the feasible during the evolutionary search procedure.

4.2 The probability-based initialization scheme

In MOEAs, the initial population generally has a great impact on the optimization performance. Unfortunately, the problem concerned in the paper is highly constrained, and based on BLS-IR, infeasible solutions dominate the search space. A random initial population is very likely to result into a deteriorated optimization performance. Hence, a probability-based initialization (PBI) scheme is devised to guarantee that the proposed MOEA begins with a set of feasible individuals with high level of diversity.

In the literature, to deal with such problem, Kim et al. insert an all-one individual into the initial population to ensure that the search starts with at least one feasible solution [10]. However, such method is not effective for MOPs, as MOEAs require higher level of population diversity than single-objective EAs. Our previous work investigates the estimated distribution of feasible solutions over the entire search space [29]. It was found that the majority of feasible solutions are closer to the all-one individual. Based on this finding, a smart initialization scheme is proposed to generate an individual pool of multiple feasible individuals based on the all-one individual.

Nevertheless, such scheme leads to an initial population of highly similar individuals, which seriously harms the population diversity.

We introduce the concept of PV in PBIL to generate the initial population. The distribution of feasible individuals in the search space is estimated to extract statistical information. Instead of setting each value in the PV to 0.5, we set a larger probability at each position of PV, to generate feasible individuals with a higher probability. Fig.5 shows the pseudo-code of the PBI scheme. This is in compliance with the finding above, i.e. an individual similar to the all-one individual is more likely to be feasible.

The PBI scheme generates an initial population with feasible and diversified individuals. As long as P_{init} is not set too close to 1, the initial population could maintain a certain degree of diversity. A smaller P_{init} is more likely to gain a more diversified initial population, which is at the expense of longer computational time. Since diversity is extremely important for MOEAs, it is worth compromising the computational cost.

4.3 A problem-specific population updating rule

In the original MOEA/D, a better individual replaces not only the one associated with the corresponding SOSP, but also those associated with neighboring SOSPs. As the problem concerned in the paper is highly complicated and constrained, feasible individuals only account for a very small proportion of the population [29]. In addition, the majority of feasible individuals are close to the all-one individual. If we adopt the original population updating rule, where better individual is used to update every neighboring SOSP, similar individuals will rapidly spread and dominate the population, causing serious prematurity and deteriorated optimization performance.

To overcome the above problem, this paper proposes a problem-specific population updating (PSPU) rule, where, instead of multiple SOSPs, only a single SOSP is updated by the newly generated promising individual. Let the *i*-th SOSP denoted by SOSP(*i*), where i = 1, …, N. Let SOSPs(*i*) be the set of the neighbors of SOSP(*i*) including itself, where SOSPs(*i*) = {SOSP(*i*(1)),…, SOSP(*i*(*W*))}, $i(1),\ldots, i(W)\in\varphi(i)$ (see Subsection 3.1 for details). For an arbitrary SOSP(*i*), an individual y is generated after reproduction and it replaces the individual of a neighboring SOSP with the most significant fitness improvement. Note that if the newly generated individual is not better than the current individuals, it is discarded.

The fitness improvement of SOSP(*i*), $\Delta_{SOSP(i)}$, and the most significant improvement regarding the fitness among SOSPs(*i*), Δ_{max}, are defined in Eq. (9) and Eq. (10), respectively.

$$\Delta_{\mathrm{SOSP}(i)} = g(x_i|\lambda^i,\mathrm{z}) - g(y|\lambda^i,\mathrm{z}) \tag{9}$$

$$\Delta_{\max} = \max \Delta_{\mathrm{SOSP}(j)},\ j\in\varphi(i) \tag{10}$$

where, $\varphi(i)$ contains the indexes of all SOSPs in SOSPs(*i*).

Compared with the original population updating rule in MOEA/D, the proposed PSPU rule defines that a newly generated individual updates the most appropriate SOSP only. Hence, the search is guided to explore promising areas in the search space while maintaining a diversified population. With the proposed rule, MOEA/D-PBIL gains better performance as observed in Subsection 5.4.

4.4 A hybridized reproduction scheme

When designing MOEAs, exploration and exploitation at different search stages should be carefully considered to support effective search over the vast search space. Traditional EA recombination operators, e.g. crossover, recombine at least two individuals selected from the population, making use of the local information only. They perform well at the beginning of the evolution, but get worse due to gradual loss of population diversity, leading to a deteriorated global search performance. PBIL manipulates a PV and generates new individuals by sampling from it. By making use of the global and historical information, promising regions can be explored in parallel and new regions also have chance to be discovered. An effective global exploration is obtained by the intrinsic memory of PV. The recombination of PBIL can thus act as a complement to the traditional EA recombination.

To achieve a balanced global exploration and local exploitation, MOEA/D-PBIL utilizes a hybridized reproduction (HR) scheme that adopts the genetic operators of MOEA/D and the probabilistic sampling operators of PBIL at different stages of the evolution. By controlling the proportion of the offspring produced by MOEA/D and those by PBIL, MOEA/D-PBIL aims at striking a balanced global exploration and local exploitation. To be specific, at the early stage of the evolution, MOEA/D reproduction is selected with a higher probability, which helps explore the search space; at the middle stage of the evolution, both reproductions incur with similar probabilities; at the late stage, PBIL-reproduction is more likely to be selected for concentrating on promising areas in the search space.

According to the HR scheme, we need a controlling parameter CP(k) for generation k to determine how many individuals are generated by PBIL-reproduction. We find that the cumulative distribution function (CDF) of the Cauchy distribution can be adapted for controlling the PBIL offspring proportion, since the CDF curve grows gradually and smoothly from a value close to 0 to a value close to 1 [57]. CP(k) is defined in Eq. (11). This parameter determines which reproduction method is used to generate a new individual. For the i-th SOSP, if an uniformly distributed random number $rand \leq$ CP(k), then PBIL-reproduction is chosen to produce offspring; otherwise, MOEA/D-reproduction is chosen. If PBIL-reproduction cannot produce a feasible individual after a number of attempts, especially when building up the probabilistic blocks for the PV in the early stage of evolution, MOEA/D-reproduction is used instead.

$$\mathrm{CP}(k) = \frac{1}{\pi}\arctan(\frac{k - K/2}{H}) + \frac{1}{2} \tag{11}$$

where, parameter H is a predefined value governing the steepness of the CP curve and parameter K is a predefined number of generations.

Fig.6 illustrates an example curve of CP, where K is set to 200 and H is set to 4, 11, and 18, respectively. A smaller H leads to a deeper slope (in the paper, H is fixed at 11). It is clear that at early stage of evolution (k = 1 ~ 50), MOEA/D-reproduction is more likely to be chosen; in the middle stage (k = 51 ~ 150), the probability of selecting PBIL-reproduction gradually increases and becomes higher than that of MOEA/D-reproduction after k = 100; at last stage of evolution, individuals generated by PBIL-reproduction dominate the population. Using this controlling parameter, a balanced global exploration and local exploitation is obtained, leading to a decent performance as seen in Subsection 5.5.

Fig.7 illustrates the procedure of the proposed HR scheme at generation k. The HR scheme decides the percentage of individuals that each reproduction method generates at a certain generation. Let the PV associated with the i-th SOSP at generation k denoted by $\mathbf{P}^{(i,k)}$. Variable *attempts* is a counter that records how many times $\mathbf{P}^{(i,k)}$ has been sampled before a feasible individual appears. Variable '*isSuccess*' records the state whether PBIL-reproduction successfully produces a feasible individual. In the early stage of evolution, $\mathbf{P}^{(i,k)}$ focuses on learning probabilistic features from promising samples. During this period, it is very likely that sampling $\mathbf{P}^{(i,k)}$ only results into infeasible individuals. The initial state of '*isSuccess*' is set to false. Constant Θ stands for the maximum number of attempts tried when sampling $\mathbf{P}^{(i,k)}$. When PBIL-reproduction is chosen, $\mathbf{P}^{(i,k)}$ is repeatedly sampled. This procedure stops when either a feasible individual appears or Θ attempts have been tried. If no feasible individual is generated, we use MOEA/D-reproduction to produce a new individual.

The HR scheme has a significant advantage, namely helping balance the global exploration and the local exploitation during the evolution. By incorporating PBIL reproduction method, the HR scheme is able to enhance the global exploitation ability of the proposed MOEA/D-PBIL. This helps provide a balanced performance between global exploration and local exploitation during all stages of the evolution, which is in favor of gaining an excellent optimization performance.

4.5 The overall procedure of MOEA/D-PBIL

The proposed MOEA/D-PBIL is based on the basic framework of MOEA/D (see Subsection 3.1). Let k be the current generation of evolution. The following gives the whole evolutionary procedure.

Input:

— the MOP with m objectives and individual length L; the stopping criteria; the population size N; the number of neighbors W; the N weight vectors $\boldsymbol{\lambda}^1, \ldots, \boldsymbol{\lambda}^N$;

```
   the crossover rate $p_c$; the mutation rate $p_m$                      // Subsection 3.1
－ the learning rate $\alpha$, the probability shifting $\sigma$           // Subsection 3.2
－ the probability for the PBI scheme $P_{init}$                          // Subsection 4.2
－ the predefined number of generations $K$; the predefined value for smoothness $H$ in
   CP($k$); the predefined number of attempts $\Theta$                    // Subsection 4.4

MOEA/D-PBIL Procedure:
Initialization
1.  Set EP = $\Phi$ and $k$ = 0.
2.  Calculate $\varphi(i)$ neighbors for SOSP($i$), $i = 1, \ldots, N$.             // Subsection 3.1
3.  Generate a population $x_1, \ldots, x_n$ by the PBI scheme.                // Subsection 4.2
4.  Initialize $z = (z_1, \ldots, z_m)$.
5.  Initialize $\mathbf{P}^{(i,k)} = (P_1^k, \ldots, P_L^k)$, $i = 1, \ldots, N$.               // Subsection 3.2
Repeat
6.  For $i$ = 1 to $N$ Do
7.      Reproduction: Produce individual $y$ by the HR scheme.    // Subsection 4.4
8.      Update of z: For each $j \in \{1, \ldots, m\}$, if $f_j(y) \le z_j$, set $z_j = f_j(y)$.
9.      Update of population: The PSPU rule is used.              // Subsection 4.3
10.     Update of PV: Update $\mathbf{P}^{(i,k)}$ by Eq. (5) and Eq. (6).          // Subsection 3.2
11.     Update of EP: Remove those dominated by $y$ from EP and add $y$ to EP if it is
        not dominated by anyone in EP.                            // Subsection 3.1
Termination
12. If stopping criterion is satisfied, stop and output EP.
```

In Step 3, the PBI scheme is used to generate an initial population, where PV $\mathbf{P}^{(init)}$ is repeatedly sampled in order to guarantee that every individual in the population is feasible. This provides the proposed algorithm a set of promising and diversified individuals to begin with. PBIL reproduction method is integrated into the MOEA/D framework (see Section 4.4). In MOEA/D-PBIL, each SOSP is associated with a PV, e.g. $\mathbf{P}^{(i,k)} = (P_1^k, \ldots, P_L^k)$. In Step 5, each PV is initialized as (0.5, 0.5, …, 0.5), where value '0.5' is the probability of generating '1' at that position. In Step 7, a new individual is produced by the MOEA/D- or PBIL- reproduction. In Step 9, the PSPU rule first calculates $\Delta_{\max}$ according to Eq. (7) and Eq. (8). For $j \in \varphi(i)$, if $\Delta_{\mathrm{SOSP}(j)} = \Delta_{\max} > 0$, then set $x_j = y$ and $f_u(x_j) = f_u(y)$, $u = 1, \ldots, m$, where $f_u(x)$ is the u-th objective value. No matter whether the PBIL reproduction method is chosen, $\mathbf{P}^{(i,k)}$ is consistently updated at each generation, where $i = 1, \ldots, N$. Step 10 defines this procedure. In Step 12, the termination condition is that the algorithm evolves a predefined number of generations.

The learning rate α defines how quickly PV learns from the best individual, and has a great impact on the convergence of PV [55]. In MOEA/D-PBIL, α is adaptively changed during the evolutionary process. At the beginning of evolution, as the quality of individuals is generally low, a small α is used so that PVs learn from promising individuals; during the evolution, the value of α is increased gradually until reaching to a maximal threshold (0.1 in this paper), which to some extent helps provide a fine local exploitation.

It is a centralized approach based on the global view of network situation. When there is a NCM request (with a source node, a set of receivers, and an expected data rate), the proposed MOEA is launched to approximate the Pareto-optimal front according to the current states/parameters of nodes/links in the network and in the end output a set of compromised solutions (i.e. nondominated solutions), where each is a feasible routing plan for the request. These solutions are useful for evaluating the overall performance of NCM with multiple criteria, including the total link cost, the total coding cost, and the end-to-end delay.

Decision makers (DMs) of the network, e.g. network operators and service providers, are possibly interested in this research since they need to compromise between benefits and demands. For example, they would like to reduce the bandwidth and computational costs while end users only care about the user experience. DMs know well about how the network runs. But, NCM is a new technological concept to them. They may not clearly know its cost and benefit. So, this research helps DMs gain an initial idea about if NCM is applicable to the network and if yes, how it could be deployed in the future.

As aforementioned, the proposed approach output is a set of nondominated solutions. DMs of the network are professionals full of knowledge and experience on how to efficiently operate the network. We reasonably assume they are capable of assigning appropriate weights to the three objectives. After normalization, all objectives can be aggregated into a single objective based on weighted sum. Hence, the best one is selected to support the NCM data transmission. It is noted that the weights could be either fixed or tunable according the current status of the network. By this way, DMs can configure the network in terms of the weight setting.

5 Performance evaluation

This section studies the effectiveness of the three performance-enhancing schemes, i.e. the PBI scheme, the PSPU rule and the HR scheme, respectively. Then, we evaluate the overall performance of MOEA/D-PBIL by comparing it against a number of state-of-the-art MOEAs, including NSGA-II [31] and SPEA2 [32].

5.1 Test instances

Fourteen widely used benchmark instances are considered in this paper, including ten randomly generated networks (Rnd-1 to Rnd-10, with network size from 20 to 60, see details in [29]) and four fixed networks (7copy, 7hybrid, 15copy and 15hybrid, see details in [16]). The associated parameters of the instances are given in Table 1.

In the paper, for an arbitrary $e \in E$, its link cost $c_{link}(e)$ and propagation delay are uniformly distributed in the range of [5, 15] and [2*ms*, 10*ms*], respectively. The coding cost C_{code} is the number of coding links in the obtained $G_{NCM}(s,T)$. We assume any coding operation consumes the same amount of processing time, i.e. 1*ms*. The predefined number of generations for all algorithms for comparison is set to 200. All experiments were run on a Windows 8 OS computer with Intel(R) Core(TM) i7-3740QM CPU 2.7 GHz and 8 GB RAM. The results are obtained by running each algorithm 30 times, from which the statistics are collected and analyzed [58].

5.2 Performance Measures

To thoroughly evaluate the performance of the proposed algorithm, we employ five widely recognized performance measuring metrics throughout the experiments.

Let PF_{ref} be a reference set of solutions well approximating the true PF, and PF_{known} be the set of nondominated solutions obtained by an algorithm. Note that we may not know the true PF for highly complex multi-objective optimization problems, including the problem concerned in this work, so we combine the best-so-far solutions obtained by all algorithms after all runs and select the nondominated solutions as the reference set. This method has been widely adopted when evaluating multi-objective algorithms in the literature.

— **Inverted generational distance (IGD) [30]**: IGD is defined as the average distance from each point v in PF_{ref} to its nearest counterpart in PF_{known}, as follows:

$$IGD = \frac{\sum_{v \in PF_{ref}} d(v, PF_{known})}{|PF_{ref}|} \qquad (12)$$

where $d(v, PF_{known})$ is the Euclidean distance (in the objective space) between solution v in PF_{ref} and its nearest solution in PF_{known} and $|PF_{ref}|$ is the number of solutions in PF_{ref}. IGD measures the convergence and diversity of an obtained nondominated solution set. This metric is commonly used and a lower IGD indicates a better overall performance of an algorithm.

— **Generational distance (GD) [59]**: GD measures the average distance from each point v in PF_{known} to its nearest counterpart in PF_{ref}, as defined below:

$$GD = \sqrt{\frac{\sum_{v \in PF_{known}} d(v, PF_{ref})}{|PF_{known}|}} \qquad (13)$$

where $d(v, PF_{ref})$ is the Euclidean distance between v in PF_{known} and its nearest point in PF_{ref}. This metric is used to measure how closely PF_{known} converges to PF_{ref}. A smaller GD indicates the obtained PF is closer to the true PF.

— **Maximum spread (MS) [59]**: MS reflects how well the true PF is covered by the points in PF_{known} through the hyperboxes formed by the extreme function values observed in PF_{ref} and PF_{known}, as shown in Eq. (14).

$$MS = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(\frac{\min(f_i^{\max}, F_i^{\max}) - \max(f_i^{\min}, F_i^{\min})}{F_i^{\max} - F_i^{\min}}\right)^2} \qquad (14)$$

where m is the number of objectives; $f_i^{\max}$ and $f_i^{\min}$ are the maximum and minimum values of the i-th objective in PF_{known}, respectively; $F_i^{\max}$ and $F_i^{\min}$ are the maximum and minimum values of the i-th objective in PF_{ref}, respectively. A larger MS shows the obtained PF has a better spread.

- **Student's *t*-test [57]**: This test is to compare two algorithms in terms of the IGD values obtained in 30 runs. In this paper, two-tailed *t*-test with 58 degrees of freedom at a 0.05 level of significance is used. Given two algorithms, e.g. Alg.1 and Alg.2, the *t*-test result Alg.1↔Alg.2 indicates if the performance of Alg.1 is significantly better than, significantly worse than, or statistically equivalent to that of Alg.2, respectively.
- **Average Computational Time (ACT):** ACT is the average running time consumed by an algorithm over 30 runs. This metric is a direct indicator of the computational complexity of an algorithm.

5.3 The effectiveness of the PBI scheme

In general, an initial population should contain a considerable amount of diverse and feasible individuals. A PBI scheme is proposed (see Subsection 4.2) to generate a set of feasible initial individuals, by repeatedly sampling an initial PV $\mathbf{P}^{(init)} = (P_{init}, P_{init}, \ldots, P_{init})$ until a predefined number of feasible individuals are created. The value of P_{init} is of vital importance to the performance of the PBI scheme. Three different settings are compared in the proposed scheme. Besides, the PBI scheme is also compared with two existing initialization schemes, i.e. Kim's method [10] and Xing's method [29], as listed below.

- **Kim's method** [10]: the initial population is randomly generated. An all-one individual is included into the population to ensure the search start with a feasible point. It is widely used in the network coding resource minimization problem.
- **Xing's method** [29]: one-bit mutation is performed on the all-one individual and its variants to produce a set of feasible individuals that are very closely distributed around the all-one individual. This method has been adopted in NCM MOP.
- **PBI**: the proposed initialization scheme. Three settings, i.e. P_{init} = 0.7, 0.8 and 0.9, are tested. For simplicity, we represent them as PBI(0.7), PBI(0.8) and PBI(0.9), respectively. A larger P_{init} leads to a higher probability of generating '1' at each position. This is in compliance with the research finding in [29], where individuals closer to the all-one individual are more likely to be feasible.

As aforementioned, IGD reflects the overall performance of an algorithm regarding the quality of the obtained PF_{known}. Hence, IGD is used to evaluate the initial population. We compare the three initialization methods using six selected test instances, including 2 small instances (Rnd-1 and Rnd-2), 2 medium-sized instances (Rnd5 and Rnd7), and 2 large instances (Rnd-8 and 7-copy). Fig.8 illustrates the IGD values and their corresponding running time of the three methods.

It is clearly seen that the PBI scheme performs significantly better than the other methods in terms of IGD. Kim's method provides an initial population with at least a feasible individual. However, infeasible individuals still account for the majority of the population [29], thus the individuals in the objective domain are far away from PF_{ref}. Xing's method produces a feasible population, however without a decent diversification, i.e. all individuals are too close to the all-one individual. The proposed PBI scheme considers not only the feasibility but also the diversity of the population and thus obtains the best performance.

Kim's and Xing's methods are simple, thus both consume a smaller amount of time in all instances. On the other hand, the computational cost of the PBI scheme has a wider spread in different instances. The smallest time cost of PBI is comparable to that of Kim's and Xing's methods.

With respect to different settings of P_{init} of the PBI scheme, it is easily seen that PBI(0.7) and PBI(0.8) are better than PBI(0.9) in all instances. Sampling a $\mathbf{P}^{(init)}$ with larger P_{init}, tends to produce more feasible individuals. Therefore, to produce the same number of individuals, smaller P_{init} takes more time. On the contrary, however, it is more likely to form a diversified population (according to the studies in [55]), thus PBI(0.7) and PBI(0.8) have better IGD values than PBI(0.9). Considering not only the quality of the population, but also the time efficiency, we hereafter set $P_{init} = 0.9$ in all experiments.

To further evaluate the superiority of the PBI scheme, we run MOEA/D with the three different initialization schemes, namely A1 with Kim's method, A2 with Xing's method and A3 with PBI(0.9), on the above six selected instances. Results of Student's *t*-test are shown in Table 2. Symbols '+' and '~' in column Alg.1↔ Alg.2 indicate that algorithm Alg.1 is significantly better than and statistically equivalent to algorithm Alg.2, respectively, in terms of IGD. It is clear that A3 performs better than A2 and A1 in all instances. This also demonstrates that by providing the algorithm with a diversified and feasible population, the PBI scheme helps to guide the search towards the true PF.

5.4 The effectiveness of the problem-specific population updating rule

To evaluate the effectiveness of the PSPU rule (Subsection 4.3), we compare two MOEAs regarding the optimization results obtained, as listed below.

- **A3**: MOEA/D [30] with the PBI method, where the original population updating rule is utilized.
- **A4**: A3 with the PSPU rule.

The results of IGD, GD, MS and the t-test based on IGD values are reported in Table 3. A4 clearly outperforms A3 regarding all measures in almost all instances, indicating the effectiveness of the PSPU rule. The proposed rule is in compliance with the nature of the MOP being tackled and it deals with the issue that the search space is dominated by infeasible individuals. Feasible individuals are difficult to generate during the evolution. When promising individuals appear, if no limitation is defined to update the population, their genes could be rapidly spread over the population within a few generations. The population diversity would then be lost quickly. The PSPU rule limits the number of SOSPs to be updated and it helps maintain a certain level of diversification. Regarding the result of *t*-test, A4 performs at least no worse, but often better than A3 (in 5/6 instances). This confirms the contribution of the PSPU rule to the proposed algorithm.

5.5 The effectiveness of the HR scheme

The HR scheme in Subsection 4.4 uses two reproduction methods, from MOEA/D and PBIL, to strike a balance between global exploration and local exploitation at different stages of evolution. The effectiveness of this scheme is evaluated by comparing the following two MOEAs.

- **A4**: Original MOEA/D with the PBI method and the PSPU rule.
- **A5**: A4 with the HR scheme, namely the proposed MOEA/D-PBIL.

The results of IGD, GD, MS and the t-test based on IGD are shown in Table 4. A5 outperforms A4 in all instances regarding IGD, GD and MS. This is because the HR scheme provides A5 with a decent global exploration and local exploitation during the evolution. On the other hand, with the evolution continuing, the global exploration ability gradually decreases in the original MOEA/D, leading to local optima. In terms of the IGD *t*-test, A5 performs significantly better than A4 in 4 instances while the two achieve statistically equivalent performance in 2 instances. This also indicates the HR scheme improves the performance of the proposed MOEA/D-PBIL.

5.6 The overall performance evaluation

Finally, we thoroughly investigate the proposed MOEA/D-PBIL by comparing the following seven MOEAs.

- **NSGA-II**: As one of the classical MOEAs originally proposed by Deb et al. [31], NSGA-II is featured with three significant features, a fast nondominated sorting approach, an elitism approach and a parameter-free diversity preservation scheme. We set the population size N = 100, the crossover rate p_c = 0.9 and the mutation rate p_m = 1/L, where L is the individual length.
- **NSGA-II-Xing**: With two improvements, i.e. Xing's initialization method (in Subsection 5.3) and an individual delegation scheme in favor of diversification, NSGA-II-Xing is able to gain promising optimization performance for the bi-objective MOP with network coding in our previous work [29].
- **SPEA2**: The strength Pareto evolutionary algorithm 2 is another widely recognized MOEA [32]. We denote the archive size by N_{arc} and set N_{arc} = N = 100, p_c = 0.9 and p_m = 1/L, respectively.
- **MOPSO**: The multiobjective algorithm based on particle swarm optimization is

well-known and widely used for performance comparison [60]. A population of 100 particles is maintained. We set p_m = 1/L and 30 divisions for the adaptive grid.

- **MOPBIL1**: the multiobejctive PBIL proposed by Kim et al. has been reported to outperform a number of GA-based MOEAs when solving MOPs in the context of the robot soccer system[61]. When updating the *i*-th PV, a solution randomly selected from the archive is used. Let the number of PVs, the learning rate, and the amount of shift in the mutation be denoted by n_{PV}, α, and σ, respectively. We set N = 100, N_{arc} = 50, n_{PV} = 100, α = 0.15, p_m = 0.02, and σ = 0.2.
- **MOPBIL2**: the first multiobejctive PBIL presented by Bureerat and Sriworamas [62]. the *i*-th PV is updated by 5 solutions randomly selected from the archive. We set N = 100, N_{arc} = 50, n_{PV} = 100, α = 0.15, p_m = 0.02, and σ = 0.2.
- **MOEA/D-PBIL**: The improved MOEA/D proposed in this paper. We set N = 100, p_{init} = 0.9, r = 11, p_c = 0.9 and p_m = 1/L, respectively.

Results of IGD, GD, MS are collected in Table 5, 6, and 7, respectively. In terms of IGD and GD, MOEA/D-PBIL performs the best, obtaining the minimum IGD values in all 14 instances and the minimum GD values in 12 instances (except Rnd-4 and Rnd-8 in Table 8). The data indicates that the nondominated solutions obtained by MOEA/D-PBIL are the closest to the true PF. MOEA/D-PBIL also achieves the best spread to the true PF, since it achieves the highest MS in 11 instances (except Rnd-5, Rnd-9 and 7hybrid in Table 9). According to IGD, GD and MS, MOEA/D-PBIL gains the best performance when considering all instances due to the PBI scheme, PSPU rule, and HR scheme, which leads to a balanced trade-off between the global exploration and local exploitation, achieving better diversity and convergence at the same time. Student's *t*-test results in Table 8 indicate that MOEA/D-PBIL is the best algorithm among all algorithms, performing no worse, and usually better than the others in most of the instances.

As mentioned before, IGD is a distance metric that is qualified to unveil the overall performance of an MOEA. With the evolution continuing, IGD value of an MOEA gradually decreases for a minimization problem. So, the IGD values over generations, to a certain extent, show the convergence trend of each algorithm. We plot IGD value obtained by each MOEA at each generation on two selected instances, namely Rnd-9 and 7copy, in Fig.9. All algorithms have a fast convergence in the first 20 generations and after that each algorithm converges quite slowly until 100-th generation.

The results of ACT in Table 9 shows that compared with the others, the proposed MOEA/D-PBIL achieves the smallest ACTs in all instances. This is because the computational complexity of addressing multiple SOSPs simultaneously is usually much lower than solving a MOP directly [30]. Smaller ACT indicates that the corresponding algorithm is with lower computational complexity and it is less computationally expensive than the others. Running fast means responding quickly. It is a significant advantage of the proposed algorithm when considering its real world deployment.

6 Conclusion and future work

6.1 Conclusion

This paper formulates a multi-objective optimization problem in the context of multicasting with network coding, where the three objectives, namely the coding cost, link cost and the end-to-end delay are minimized simultaneously. Population-based incremental learning (PBIL) components are incorporated into the evolutionary framework, and a modified multi-objective evolutionary algorithm based on decomposition (MOEA/D-PBIL) is proposed. Three performance-enhancing schemes are developed, namely the probability-based initialization scheme, the hybridized reproduction operator and the problem-specific population updating rule. The first scheme is able to offer a feasible and diversified initial population; the second one is in favor of avoiding the prematurity effect; and the last one helps to balance the global exploration and local exploitation during the search. The proposed approach is centralized and based on the global view of network simulation. The experimental results demonstrate that with the three new schemes, the proposed MOEA/D-PBIL algorithm gains the best optimization performance in terms of performance indicators, namely the inverted generational distance, generational distance, maximum spread, *t*-test results, compared with six state-of-the-art MOEAs in the literature.

6.2 Future work

Although it is quite promising in concept, network coding based multicast (NCM) is difficult to implement in nowadays Internet. This is because NCM requires network forwarding devices, such as routers and switches, to perform complicated encoding operations, e.g. finite field calculations. But currently, typical network devices cannot afford such heavy computational overhead. The work reported in this paper is an initial attempt to apply NCM to communication networks from the point of view of theoretical studies, with pure mathematical models considered. It is not a real-network application. A number of important issues will be carefully considered, including the source synchronization, the traffic control schemes, the group membership change due to new member joining or old member leaving, more precise formulation for real problems and the determination of measurable and applicable metrics in multi-objective optimization.

In the future, we would like to build a network simulation test-bed for evaluating traditional multicast (TM) and NCM. Recently, software-defined networking (SDN) is an emerging network architecture that decouples the control plane from the data plane and allows network functions, e.g. forwarding rules, directly programmable [63, 64]. Fig.10 illustrates the SDN architecture. Different from traditional Internet architecture, SDN consists of three layers, namely, the infrastructure layer, the control layer and the application layer. Control plane is located in a centralized SDN controller. In data plane, network devices only follow the controller's instructions to process incoming packets, e.g. simply forwarding or dropping them. The southbound interface connects the control and infrastructure layers, where OpenFlow (OF) protocol is the de-facto communication standard. In addition, APIs are used to deploy business applications on the controller. Based on SDN, it is easy to test novel ideas, methods and protocols [63]. So, we will build a SDN-based network simulation test-bed for the performance evaluation of NCM and the proposed MOEA.

The test-bed is based on Mininet, an open-source network emulator that can generate a network of virtual hosts (VH), switches, links and controllers [65]. Mininet networks run standard Unix/Linux network applications and real Linux kernel and network stack. In Mininet, VHs run Linux network software and switches are OF-enabled and based on Open vSwitch (a multilayer virtual switch standard). Also, the emulator supports a number of popular controllers, such as, Ryu [66], NOX [67], Floodlight [68], etc. We plan to use Ryu as the controller since it has support for several OF versions, including OF-1.0 and OF-1.3 that have been widely supported by vendors. Table 10 shows the description of the Mininet-based simulation test-bed.

An example simulation scenario based on the topology in Fig.1(a) is illustrated in Fig.11. There are 3 VHs including source VH-*s* and receivers VH-t_1 and VH-t_2, 7 OF-enabled switches and an Ryu controller. Each VH is connected to a switch by two links. This is because two link-disjoint paths are to be constructed for the NCM data transmission. Every switch is controlled by the controller by following its instructions. The TM and NCM modules will be installed on the controller, providing multicast service for performance comparison.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China (No.61401374), the Fundamental Research Funds for the Central Universities and the Project-sponsored by SRF for ROCS, SEM, P. R. China.

References

1. L. J. Harte, Introduction to data multicasting, Althos, 2008.
2. A. Benslimane, Multimedia multicast on the internet, ISTE, 2007.
3. R. Ahlswede, N. Cai, S. Y. R. Li, R. W. Yeung, Network information flow, *IEEE Transactions on Information Theory*, 46(4): 1204-1216, 2000.
4. S. Y. R. Li, R. W. Yeung, N. Cai, Linear network coding, *IEEE Transactions on Information Theory*, 49 (2): 371-381, 2003.
5. N. Cai, R. W. Yeung, Secure network coding on a wiretap network, *IEEE Transactions on Information Theory*, 57(1): 424-435, 2011.
6. C. Fragouli, J. Y. L. Boudec, J. Widmer, Network coding: an instant primer, *Computer Communications Review*, 36(1): 63-68, 2006.
7. A. E. Kamal, 1+N Network Protection for Mesh Networks: Network Coding-Based Protection Using p-Cycles, *IEEE/ACM Transactions on Networking*, 18(1): 67-80, 2010.
8. C. Fragouli, E. Soljanin, Information flow decomposition for network coding, *IEEE Transactions on Information Theory*, 52(3): 829-848, 2006.
9. M. Langberg, A. Sprintson, J. Bruck, The encoding complexity of network coding, *IEEE Transactions on Information Theory*, 52(6): 2386-2397, 2006.
10. M. Kim, M. Médard, V. Aggarwal, U. M. O. Reilly, W. Kim, C. W. Ahn, M. Effros, Evolutionary approaches to minimizing network coding resources, in *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, pp. 1991-1999, 2007.
11. H. Xing, Y. Ji, L. Bai, Y. Sun, An improved quantum-inspired evolutionary algorithm for coding resource optimization based network coding multicast scheme, *AEU-International Journal of electronics and communications*, 64(12): 1105-1113, 2010.
12. H. Xing, R. Qu, A population based incremental learning for network coding resources minimization, *IEEE Communications Letters*, 15(7): 698-700, 2011.
13. H. Xing, R. Qu, A compact genetic algorithm for the network coding based resource minimization problem, *Applied Intelligence*, 36(4): 809-823, 2012.
14. H. N. Luong, H. T. T. Nguyen, C. W. Ahn, Entropy-based efficiency enhancement techniques for evolutionary algorithms, *Information Sciences*, 188: 100-120, 2013.
15. H. Xing, R. Qu, G. Kendall, R. Bai, A path-oriented encoding evolutionary algorithm for network coding resource minimization, *Journal of the Operational Research Society*, 65:1261-1277, 2013.
16. Z. Wang, H. Xing, T. Li, Y. Yang, R. Qu, Y. Pan, A modified ant colony optimization algorithm for network coding resource minimization, *IEEE Transactions on Evolutionary Computation*, 20(3): 325-342, 2016.
17. D. S. Lun, N. Ratnakar, M. Médard, R. Koetter, D. R. Karger, T. Ho, E. Ahmed, F. Zhao, Minimum-cost multicast over coded packet networks, *IEEE Transactions on Information Theory*, 52(6): 2608-2623, 2006.
18. T. Cui, T. Ho, Minimum cost integral network coding, in *Proc. International Symposium on Information Theory and its Applications (ISIT)*, pp. 2736-2740, 2007.
19. F. Zhao, M. Médard, D. Lun, A. Ozdaglar, Minimum-cost subgraph algorithms for static and dynamic multicasts with network coding, in *Proc. New Directions in Wireless Communications Research*, pp. 317-349, 2009.

20. A. Striegel, G. Manimaran, A survey of QoS multicasting issues, *IEEE Communications Magazine*, 40(6): 82-87, 2002.

21. H. Li, X. Liu, W. He, J. Li, W. Dou, End-to-end delay analysis in wireless network coding: A network calculus-based approach, in *Proc. International Conference on Distributed Computing Systems (ICDCS)*, pp. 47-56, 2011.

22. N. Cleju, N. Thomos, P. Frossard, Network coding node placement for delay minimization in streaming overlays, in *Proc. IEEE International Conference on Communications (ICC)*, pp. 1-5, 2010.

23. E. Drinea, L. Keller, C. Fragouli, Real-time delay with network coding and feedback, *Physical Communication*, 6:100-113, 2013.

24. S. Sorour, S. Valaee, Completion delay minimization for instantly decodable network coding with limited feedback, in *Proc. IEEE International Conference on Communications (ICC)*, pp. 1-5, 2011.

25. A. Douik, S. Sorour, M. Alouini, T. Y. Al-Naffouri, Delay reduction in lossy intermittent feedback for generalized instantly decodable network coding, in *Proc. IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 388-393, 2013.

26. W. Yeow, A. T. Hoang, C. Tham, Minimizing delay for multicast-streaming in wireless networks with network coding, in *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, pp. 190-198, 2009.

27. M. Kim, M. Médard, V. Aggarwal, V. O. Reilly, On the coding-link cost tradeoff in multicast network coding, in *Proc. International Conference for Military Communications (MILCOM)*, pp. 1-7, 2007.

28. C. W. Ahn, J. C. Yoo, Multi-objective evolutionary approach to coding-link cost trade-offs in network coding, *Electronics Letters*, 48(25): 1595-1596, 2012.

29. H. Xing, R. Qu, A nondominated sorting genetic algorithm for bi-objective network coding based multicast routing problems, *Information Sciences*, 233: 36-53, 2013.

30. Q. Zhang, H. Li, MOEA/D: A multi-objective evolutionary algorithm based on decomposition, *IEEE Transactions on Evolutionary Computation*, 11(6): 712-731, 2007.

31. K. Deb, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, 6(2): 182-197, 2002.

32. E. Ziztler, M. Laumanns, L. Thiele. Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization, *Evolutionary Methods for Design, Optimization, and Control*, pp. 95-100, 2002.

33. M. Pelikan, D. E. Goldberg, F. G. Lobo, A survey of optimization by building and using probabilistic models, *Computational optimization and applications*, 21(1): 5-20, 2002.

34. V. A. Shim, K. C. Tan, K. K. Tan, A hybrid estimation of distribution algorithm for solving the multi-objective multiple traveling salesman problem, in *Proc. IEEE Congress on Evolutionary Computation (CEC)*, pp. 1-8, 2012.

35. V. A. Shim, K. C. Tan, K. K. Tan, A hybrid adaptive evolutionary algorithm in the domination-based and decomposition-based frameworks of multi-objective optimization, in *Proc. IEEE Congress on Evolutionary Computation (CEC)*, pp. 1-8, 2012.

36. I. Giagkiozis, R. C. Purshouse, P. J. Fleming, Generalized decomposition and cross entropy methods for many-objective optimization, *Information Sciences*, 282: 363-387, 2014.

37. H. Li, Q. Zhang, Multiobjective optimization problems with complicated Pareto sets, MOEA/D

and NSGA-II, *IEEE Transactions on Evolutionary Computation*, 13(2): 284-302, 2009.

38. Y. Tan, Y. Jiao, H. Li, X. Wang, A modification to MOEA/D-DE for multiobjective optimization problems with complicated Pareto sets, *Information Sciences*, 213: 14-38, 2012.
39. R. Carvalho, R. R. Saldanha, B. N. Gomes, et al., A multi-objective evolutionary algorithm based on decomposition for optimal design of Yagi-Uda antennas, *IEEE Transactions on Magnetics*, 48(2): 803-806, 2012.
40. S. M. Venske, R. A. Gonçalves, M. R. Delgado, ADEMO/D: Multiobjective optimization by an adaptive differential evolution algorithm, *Neurocomputing*, 127: 65-77, 2014.
41. Y. Mei, K. Tang, X. Yao, Decomposition-based memetic algorithm for multiobjective capacitated arc routing problem, *IEEE Transactions on Evolutionary Computation*, 15(2): 151-165, 2011.
42. R. Shang, J. Wang, L. Jiao, et al., An improved decomposition-based memetic algorithm for multi-objective capacitated arc routing problem, *Applied Soft Computing*, 19: 343-361, 2014.
43. M. S. Zapotecas, C. A. Coello Coello, A direct local search mechanism for decomposition-based multi-objective evolutionary algorithms, in *Proc. IEEE Congress on Evolutionary Computation (CEC)*, pp. 1-8, 2012.
44. W. K. Mashwani, A. Salhi, Multiobjective memetic algorithm based on decomposition, *Applied Soft Computing*, 21: 221-243, 2014.
45. L. Ke, Q. Zhang, R. Battiti, Hybridization of decomposition and local search for multiobjective optimization, *IEEE Transactions on Cybernetics*, 44(10): 1808-1820, 2014.
46. H. Li, D. Landa-Silva, X. Gandibleux, Evolutionary multi-objective optimization algorithms with probabilistic representation based on pheromone trails, in *Proc. IEEE Congress on Evolutionary Computation (CEC)*, pp. 1-8, 2010.
47. L. Ke, Q. Zhang, R. Battiti, MOEA/D-ACO: a multiobjective evolutionary algorithm using decomposition and Ant Colony, *IEEE Transactions on Cybernetics*, 43(6): 1845-1859, 2013.
48. J. Cheng, G. Zhang, Z. Li, et al., Multi-objective ant colony optimization based on decomposition for bi-objective traveling salesman problems, *Soft Computing*, 16(4): 597-614, 2012.
49. M. N. Al, A. Petrovski, J. McCall, D 2 MOPSO: multi-objective particle swarm optimizer based on decomposition and dominance, in *Proc. Evolutionary Computation in Combinatorial Optimization*, Springer, pp. 75-86, 2012.
50. H. Li, D. Landa-Silva, An adaptive evolutionary multi-objective approach based on simulated annealing, *Evolutionary Computation*, 19(4): 561-595, 2011.
51. M. Nasir, A. K. Mondal, S. Sengupta, et al., An improved multiobjective evolutionary algorithm based on decomposition with fuzzy dominance, in *Proc. IEEE Congress on Evolutionary Computation (CEC)*, pp. 765-772, 2011.
52. A. Zhou, Q. Zhang, G. Zhang, A multiobjective evolutionary algorithm based on decomposition and probability model, in *Proc. IEEE Congress on Evolutionary Computation (CEC)*, pp. 1-8, 2012.
53. X. Ma, F. Liu, Y. Qi, et al., MOEA/D with opposition-based learning for multiobjective optimization problem, *Neurocomputing*, 146: 48-64, 2014.
54. M. A. Medina, J. M. Ramirez, A. C. Coello Coello, A novel multi-objective optimizer for handling reactive power, in *Proc. IEEE Grenoble PowerTech (POWERTECH)*, pp. 1-6, 2013.
55. S. Baluja, Population-based incremental learning: a method for integrating genetic search based function optimization and competitive learning, Technical Report CMU-CS-94-163, Carnegie Mellon University, 1994.

56. A. V. Goldberg, A new max-flow algorithm, Laboratory for Computer Science, MIT, 1985.
57. R. E. Walpole, R. H. Myers, S. L. Myers, K. Ye, Probability and statistics for engineers and scientists, Pearson Education, 2007.
58. S. Garcia, D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behavior: a case study on the CEC'2005 Special Session on Real Parameter Optimization. *Journal of Heuristics*, 15: 617-644, 2009.
59. K. C. Tan, Y. J. Yang, C. K. Goh, A distributed cooperative coevolutionary algorithm for multiobjective optimization, *IEEE Transactions on Evolutionary Computation*, 10(5): 527-549, 2006.
60. C. A. Coello Coello, M. S. Lechuga, MOPSO: A proposal for multiple objective particle swarm optimization, in *Proc. the IEEE Congress on Evolutionary Computation (CEC)*, 2: 1051-1056, 2002.
61. J. H. Kim, Y. H. Kim, Choi S H, et al., Evolutionary multi-objective optimization in robot soccer system for education, *IEEE Computational Intelligence Magazine*, 4(1): 31-41, 2009.
62. S. Bureerat, K. Sriworamas, Population-based incremental learning for multiobjective optimization, in *Proc. Soft Computing in Industrial Applications*, Springer, pp. 223-232, 2007.
63. W. Xia, Y. Wen, C.H. Foh, D. Niyato, H. Xie, A survey on software-defined networking, *IEEE Communication Surveys & Tutorials*, 17(1): 27-51, 2015.
64. D. Kreutz, F.M.V. Ramos, P.E. Veríssimo, C.E. Rothenberg, S. Azodolmolky, S. Uhlig, Software-defined networking: a comprehensive survey, *Proceedings of the IEEE*, 103(1): 15-76, 2015.
65. B. Lantz, B. Heller, and N. McKeown, A network in a laptop: rapid prototyping for software-defined networks, in *Proc. 9th ACM SIGCOMM Workshop HotNets-IX*, pp. 1-6, 2010.
66. Ryu. [Online]. Available: http://osrg.github.com/ryu/
67. NOX. [Online]. Available: http://www.noxrepo.org/
68. Floodlight. [Online]. Available: http://www.projectfloodlight.org/

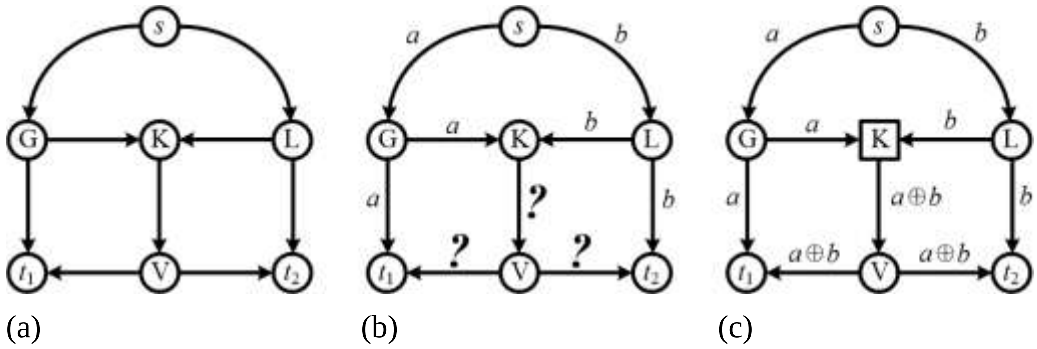


Fig.1 An example multicast scenario. (a) Network topology. (b) Traditional routing. (c) Network coding.

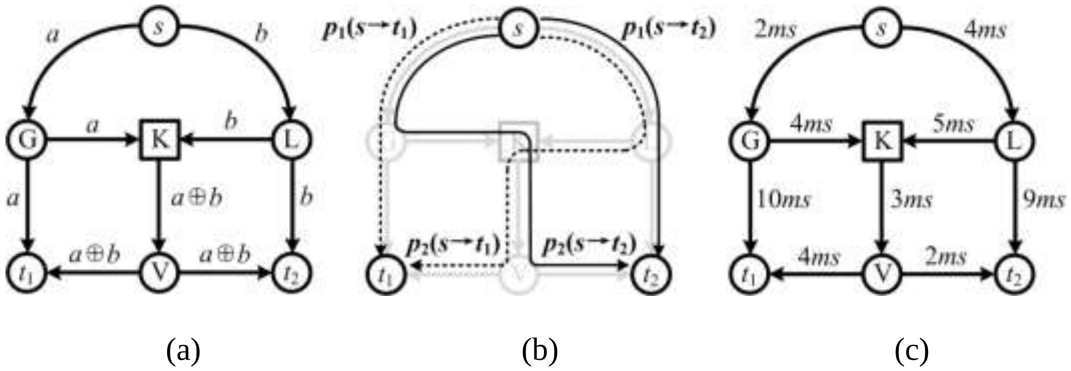
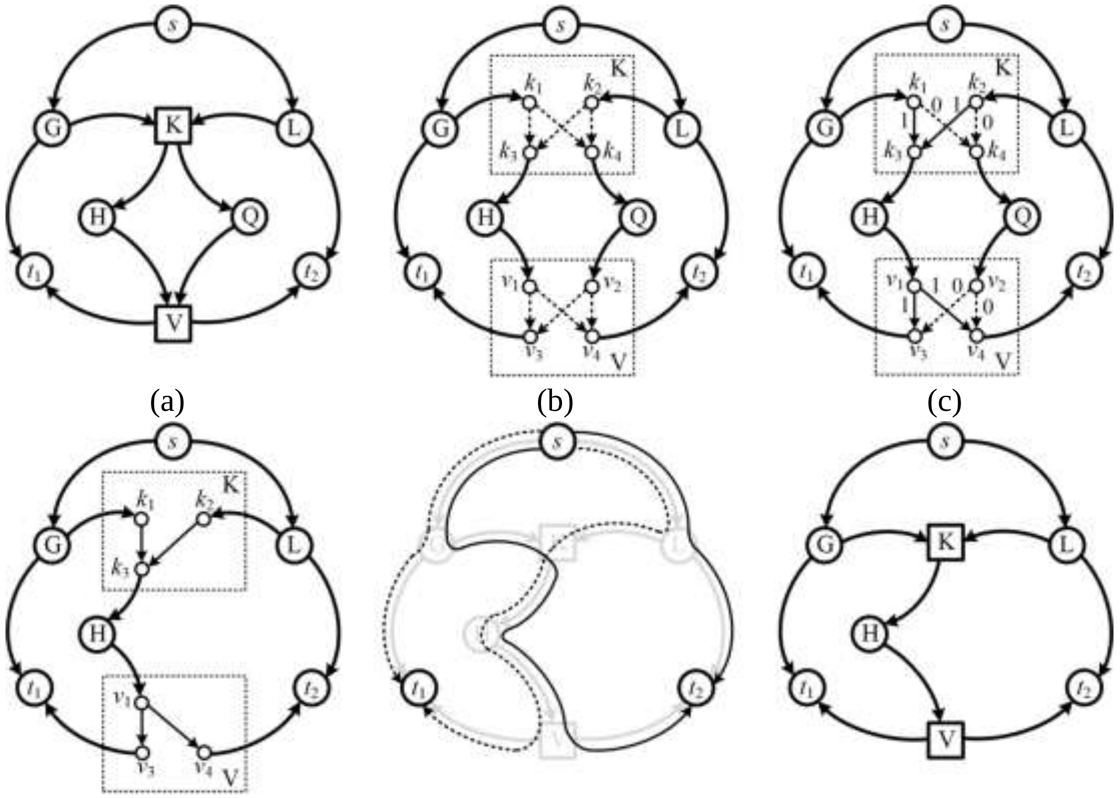


Fig.2 An example NCM scenario. (a) Data delivery. (b) The NCM subgraph. (c) The propagation delays over links.

```
Initialization
Set $k = 0$;
Set $P_l^k = 0.5$, $l = 1 \ldots L$. So $\mathbf{P}^{(k)}$ is initialized as (0.5,…,0.5);
Sampling a set $\mathbf{S}^{(k)}$ of $N$ individuals from $\mathbf{P}^{(k)}$ and find the best sample and $\mathbf{B}^{(k)}$;
Repeat
Set $k = k + 1$;
Find the best sample $\mathbf{B}^{(k)}$ from $\mathbf{B}^{(k\text{-}1)}\cup\mathbf{S}^{(k\text{-}1)}$;
Update $\mathbf{P}^{(k)}$ by Eq. (5);
Mutate $\mathbf{P}^{(k)}$ by Eq. (6);
Sampling a set $\mathbf{S}^{(k)}$ of $N$ individuals from $\mathbf{P}^{(k)}$;
Termination
Until stopping criteria is satisfied, output $\mathbf{B}^{(k)}$.
```

Fig.3 Pseudo-code of the original PBIL [55].



(a) (b) (c)

(d) (e) (f)

Fig.4 An example of the individual representation. (a) The original graph *G*. (b) The decomposed graph *G'*. (c) The link states based on '11001010'. (d) The corresponding graph G_D. (e) The paths found. (f) The constructed NCM subgraph.

```
Set initial population $set_{init}$ = ∅;
Set probability vector $\mathbf{P}^{(init)} = (P_{init}, P_{init}, \ldots, P_{init})$;
While | $set_{init}$ | < $N$
    Gerenate a new individual $x$ by sampling $\mathbf{P}^{(init)}$ once;
    If $x$ is feasible then
        Place $x$ in $set_{init}$;
Output population $set_{init}$.
```

Fig.5 Pseudo-code of the PBI scheme.

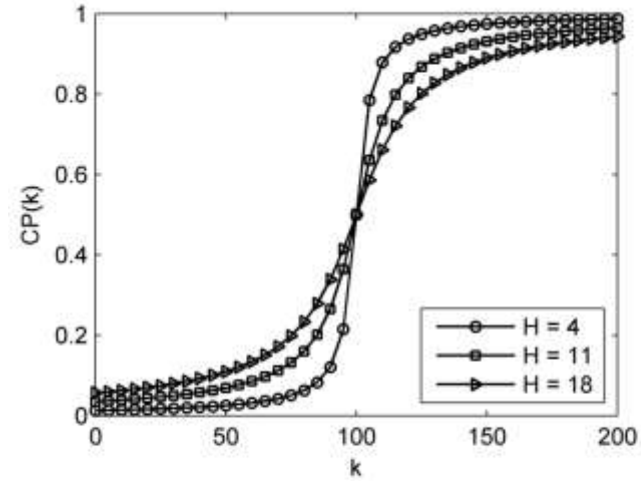
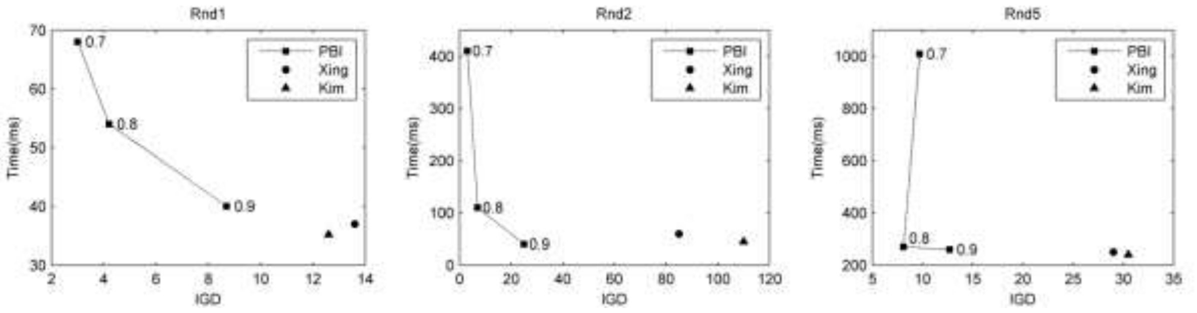


Fig.6 Example CP with K = 200 and different values of H.

```
Calculate the value of the controlling parameter CP($k$) // Eq. (9)
For $i$ = 1 to $N$ Do                                  // For the $i$-th SOSP
    Generate a random number rand;                     // uniformly distributed in [0,1]
    If rand $\leq$ CP($k$) Then Do                     // PBIL-reproduction is used
        Set attempts = 0 && isSuccess = 'false';
        While attempts < $\Theta$ Do                   // Attempts a number of times
            Set attempts = attempts + 1;
            Sample an individual $y$ from PV $\mathbf{P}^{(i,k)}$ ;   // Subsection 3.2
            If $y$ is feasible Then Do
                Set isSuccess = 'true'; Break;
        If isSuccess == 'false' Then Do                // MOEA/D-reproduction is used
            Generate an individual $y$ by crossover and mutation;   // Subsection 3.1
    Else                                               // MOEA/D-reproduction is used
        Generate an individual $y$ by crossover and mutation;
Output the offspring population.
```

Fig.7 Procedure of the HR scheme at generation k.



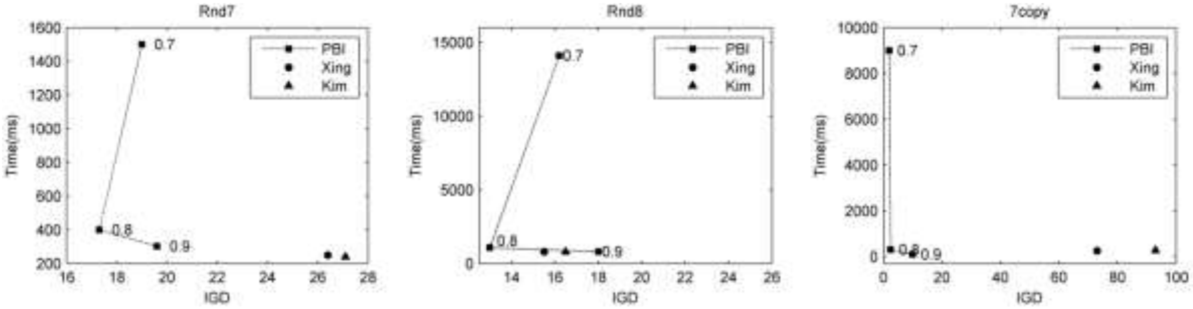


Fig.8 Comparisons among different initialization schemes.

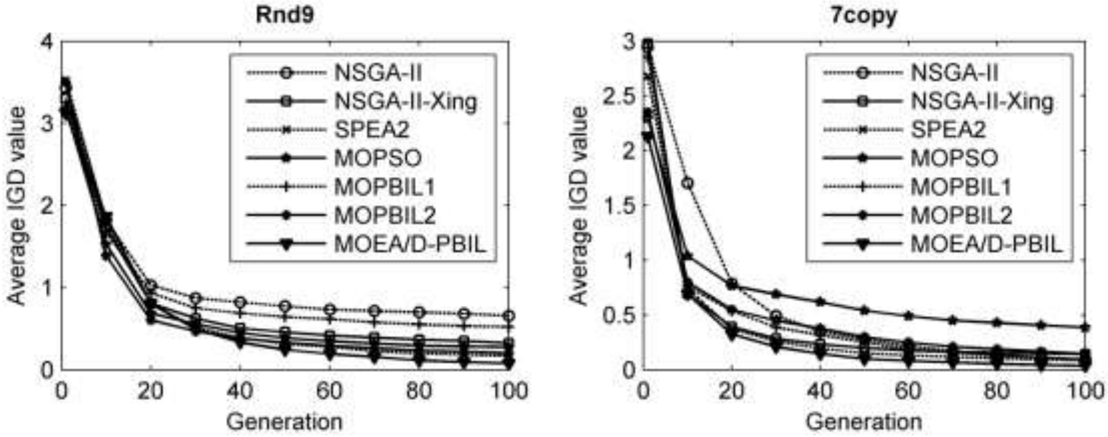


Fig.9 Convergence trends of different algorithms on two selected instances.

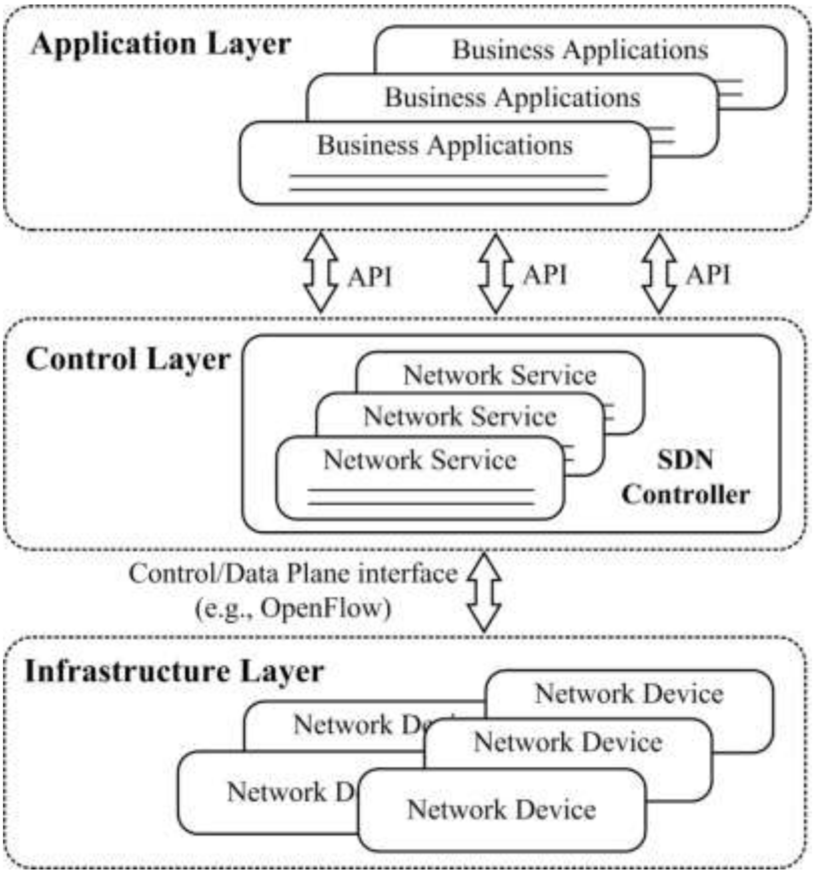


Fig.10 The Architecture of SDN.

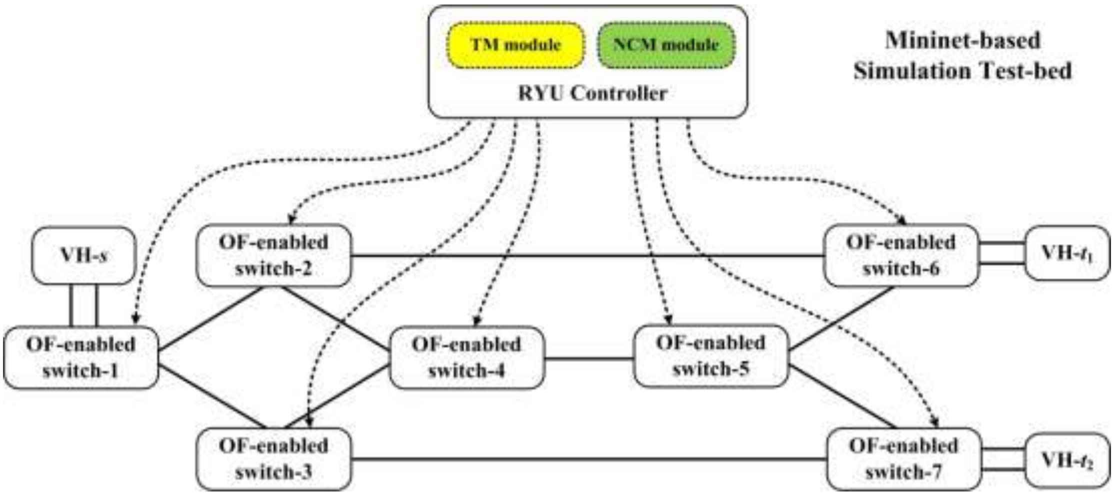


Fig.11 An example simulation scenario based on the topology in Fig.1(a).

Table 1 Test Benchmark Networks and Their Parameters [29].

Parameters	Rnd-1	Rnd-2	Rnd-3	Rnd-4	Rnd-5	Rnd-6	Rnd-7
No. of nodes	20	20	30	30	40	40	50
No. of links	37	39	60	69	78	85	101
No. of receivers	5	5	6	6	9	9	8
Data rate	3	3	3	3	3	4	3
Parameters	Rnd-8	Rnd-9	Rnd-10	7copy	15copy	7hybrid	15hybrid
No. of nodes	50	60	60	57	55	121	118
No. of links	118	150	156	84	80	180	174
No. of receivers	10	11	10	8	8	16	16
Data rate	4	5	4	2	2	2	2

Table 2 Student *t*-test results of A1, A2 and A3.

Network	Rnd-1	Rnd-2	Rnd-5	Rnd-7	Rnd-8	7copy
A3↔A1	+	+	+	+	+	+
A3↔A2	~	~	+	+	+	+

Table 3 Results of IGD, GD, MS and the IGD *t*-test (the best are in bold).

Performance Metrics		Rnd-1	Rnd-2	Rnd-5	Rnd-7	Rnd-8	7copy
IGD	A3	1.29(0.66)	1.92(1.56)	2.05(0.36)	2.84(0.66)	4.17(0.47)	0.54(0.15)
	A4	**1.11**(0.57)	**0.99**(1.32)	**1.11**(0.28)	**2.12**(0.59)	**0.72**(0.34)	**0.03**(0.02)
GD	A3	1.21(0.89)	0.42(1.18)	1.69(0.47)	3.51(1.30)	2.69(0.50)	1.14(0.21)
	A4	**0.88**(0.62)	**0.02**(0.10)	**0.66**(1.19)	**2.28**(1.25)	**0.79**(0.40)	**1.16**(0.06)
MS	A3	**1.00**(0.00)	0.83(0.23)	**0.83**(0.04)	0.82(0.03)	0.72(0.22)	0.86(0.05)
	A4	**1.00**(0.00)	**0.92**(0.18)	**0.83**(0.10)	**0.92**(0.05)	**0.83**(0.13)	**0.89**(0.03)
IGD t-test	A4↔A3	+	+	+	~	+	+

Note: in data x(y), x and y represent the mean value and standard deviation (SD) of the metric values, respectively.

Table 4 Results of IGD, GD, MS and IGD *t*-test of A4 and A5 (the best are in bold)

Performance Metrics		Rnd-1	Rnd-2	Rnd-5	Rnd-7	Rnd-8	7copy
IGD	A4	1.11(0.57)	0.99(1.32)	1.11(0.28)	2.12(0.59)	0.72(0.34)	0.03(0.02)
	A5	**0.00**(0.00)	**0.00**(0.00)	**1.07**(0.55)	**1.16**(0.43)	**0.62**(0.33)	**0.02**(0.01)
GD	A4	0.88(0.62)	0.02(0.10)	0.66(1.19)	2.28(1.25)	0.79(0.40)	1.16(0.06)
	A5	**0.00**(0.00)	**0.00**(0.00)	**0.55**(0.26)	**0.39**(0.24)	**0.72**(0.66)	**0.11**(0.01)
MS	A4	**1.00(0.00)**	0.92(0.18)	0.83(0.10)	0.92(0.05)	0.89(0.03)	0.99(0.01)
	A5	**1.00(0.00)**	**1.00(0.00)**	**0.79**(0.13)	**1.00(0.00)**	**0.92**(0.02)	**0.98**(0.01)
IGD t-test	**A5↔A4**	+	+	~	+	~	+

Note: in data x(y), x and y represent the mean value and standard deviation (SD) of the IGD values, respectively.

Table 5 Results of IGD of the seven algorithms (Best results are in bold).

Network	NSGA-II	NSGA-II-Xing	SPEA2	MOPSO	MOPBIL1	MOPBIL2	MOEA/D-PBIL
Rnd-1	**0.00**(0.00)	**0.00**(0.00)	5.13(0.67)	2.87(0.46)	0.01(0.04)	0.40(0.63)	**0.00**(0.00)

Rnd-2	**0.00**(0.00)	**0.00**(0.00)	5.69(0.55)	3.32(1.68)	0.86(1.26)	1.67(1.41)	**0.00**(0.00)
Rnd-3	**0.00**(0.00)	**0.00**(0.00)	0.87(2.44)	0.76(0.30)	**0.00**(0.00)	**0.00**(0.00)	**0.00**(0.00)
Rnd-4	1.68(0.52)	0.96(0.21)	4.01(0.25)	2.38(0.27)	0.85(0.64)	0.95(0.80)	**0.84**(0.43)
Rnd-5	1.22(1.29)	1.35(0.11)	1.56(1.10)	2.26(0.32)	1.18(0.21)	1.19(0.23)	**1.07**(0.55)
Rnd-6	**0.00**(0.00)	**0.00**(0.00)	2.04(0.05)	1.63(0.44)	1.03(0.26)	1.01(0.54)	**0.00**(0.00)
Rnd-7	1.36(0.48)	1.33(0.23)	3.35(1.61)	2.36(0.55)	1.31(0.61)	1.53(0.68)	**1.16**(0.43)
Rnd-8	1.05(0.15)	1.17(0.45)	1.56(0.46)	3.76(0.32)	1.07(0.56)	1.08(0.73)	**0.62**(0.42)
Rnd-9	0.46(0.24)	0.21(0.22)	0.12(1.34)	0.14(2.23)	0.38(0.23)	0.19(0.15)	**0.07**(0.09)
Rnd-10	0.37(0.11)	0.31(0.12)	**0.20**(0.06)	0.45(0.27)	0.49(0.11)	0.62(0.24)	**0.20**(0.05)
7copy	0.03(0.01)	0.03(0.02)	0.07(0.03)	0.36(0.08)	0.05(0.01)	0.06(0.01)	**0.02**(0.01)
15copy	2.68(0.57)	2.52(0.50)	2.19(0.60)	7.30(1.34)	3.21(2.52)	3.01(1.59)	**0.09**(0.03)
7hybrid	0.02(0.03)	0.02(0.02)	0.02(0.02)	0.43(0.14)	0.01(0.02)	0.02(0.03)	**0.01**(0.01)
15hybrid	2.24(0.54)	2.25(0.56)	2.25(0.54)	6.37(2.09)	3.42(1.89)	3.29(1.07)	**0.12**(0.10)

Table 6 Results of GD of the seven algorithms (Best results are in bold).

Network	NSGA-II	NSGA-II-Xing	SPEA2	MOPSO	MOPBIL1	MOPBIL2	MOEA/D-PBIL
Rnd-1	**0.00**(0.00)	**0.00**(0.00)	0.35(0.30)	0.65(0.30)	0.07(0.49)	0.30(0.54)	**0.00**(0.00)
Rnd-2	**0.00**(0.00)	**0.00**(0.00)	0.90(1.35)	0.90(0.35)	0.03(0.13)	0.02(0.09)	**0.00**(0.00)
Rnd-3	**0.00**(0.00)	**0.00**(0.00)	0.31(0.25)	0.19(0.22)	**0.00**(0.00)	**0.00**(0.00)	**0.00**(0.00)
Rnd-4	0.52(0.19)	0.54(0.47)	2.07(2.36)	3.03(1.02)	**0.44**(0.75)	0.69(1.03)	1.16(0.38)
Rnd-5	0.78(0.16)	0.66(0.13)	0.85(0.14)	0.97(0.16)	0.79(0.17)	0.72(0.22)	**0.55**(0.26)
Rnd-6	**0.00**(0.00)	**0.00**(0.01)	1.66(1.50)	1.72(0.45)	0.32(0.00)	0.98(0.00)	**0.00**(0.00)
Rnd-7	1.78(1.24)	1.06(0.49)	1.27(0.49)	1.25(0.64)	1.75(1.12)	1.94(1.40)	**0.39**(0.24)
Rnd-8	1.38(0.17)	1.06(0.19)	**0.43**(0.36)	2.55(0.50)	1.03(0.87)	1.06(0.97)	0.72(0.66)
Rnd-9	0.69(0.26)	0.73(0.32)	0.85(0.12)	0.83(0.16)	0.44(0.12)	0.52(0.16)	**0.33**(0.11)
Rnd-10	0.50(0.11)	0.48(0.14)	0.60(0.14)	1.12(0.38)	0.48(0.19)	0.49(0.18)	**0.31**(0.06)
7copy	0.26(0.05)	0.21(0.07)	0.39(0.16)	1.12(0.16)	0.30(0.24)	0.27(0.19)	**0.11**(0.01)
15copy	1.97(0.26)	1.73(0.31)	2.18(0.28)	3.31(0.17)	2.03(1.08)	1.45(1.23)	**0.19**(0.03)
7hybrid	1.12(0.07)	0.12(0.12)	0.35(0.13)	1.42(0.33)	0.09(0.08)	0.10(0.09)	**0.05**(0.05)
15hybrid	1.43(0.29)	1.49(0.28)	1.74(0.38)	2.68(0.15)	1.47(0.19)	1.44(0.76)	**0.17**(0.09)

Table 7 Result of MS of the seven algorithms (Best results are in bold).

Network	NSGA-II	NSGA-II-Xing	SPEA2	MOPSO	MOPBIL1	MOPBIL2	MOEA/D-PBIL
Rnd-1	**1.00**(0.00)	**1.00**(0.00)	**1.00**(0.00)	0.72(0.24)	**1.00(0.00)**	**1.00(0.00)**	**1.00(0.00)**
Rnd-2	**1.00**(0.00)	**1.00**(0.00)	**1.00**(0.00)	0.96(0.13)	0.93(0.14)	0.90(0.14)	**1.00(0.00)**
Rnd-3	**1.00**(0.00)	**1.00**(0.00)	**1.00**(0.00)	0.99(0.02)	**1.00**(0.00)	**1.00**(0.00)	**1.00(0.00)**
Rnd-4	0.87(0.02)	**0.97**(0.03)	0.81(0.02)	0.70(0.06)	0.96(0.06)	0.95(0.06)	**0.97**(0.05)
Rnd-5	0.86(0.07)	0.89(0.05)	0.86(0.02)	0.69(0.05)	**0.90**(0.04)	0.89(0.05)	0.79(0.13)
Rnd-6	**1.00**(0.00)	**1.00**(0.00)	0.76(0.10)	0.73(0.10)	0.82(0.01)	0.87(0.12)	**1.00(0.00)**
Rnd-7	0.93(0.02)	0.94(0.02)	0.94(0.02)	0.63(0.05)	0.93(0.03)	0.92(0.02)	**1.00**(0.00)
Rnd-8	0.63(0.01)	0.87(0.11)	0.80(0.06)	0.59(0.03)	0.86(0.07)	0.84(0.03)	**0.92**(0.02)
Rnd-9	0.90(0.09)	**0.97**(0.04)	**0.97**(0.02)	0.85(0.14)	0.84(0.07)	0.85(0.06)	0.96(0.03)
Rnd-10	0.95(0.03)	0.96(0.02)	0.94(0.20)	0.62(0.15)	0.91(0.03)	0.92(0.04)	**0.97**(0.02)
7copy	**0.99**(0.01)	**0.99**(0.01)	**0.99**(0.01)	0.51(0.09)	0.97(0.01)	0.98(0.01)	**0.99**(0.01)
15copy	0.49(0.11)	0.54(0.02)	0.57(0.06)	0.72(0.04)	0.48(0.09)	0.51(0.11)	**0.98**(0.01)
7hybrid	0.99(0.01)	0.99(0.01)	**1.00**(0.00)	0.75(0.03)	0.99(0.02)	0.99(0.01)	0.99(0.01)
15hybrid	0.89(0.08)	0.91(0.03)	0.93(0.03)	0.76(0.04)	0.81(0.03)	0.81(0.03)	**0.97**(0.03)

Table 8 Student's *t*-test results based on IGD of the seven algorithms.

Alg. 1 ↔ Alg. 2	Rnd-1	Rnd-2	Rnd-3	Rnd-4	Rnd-5	Rnd-6	Rnd-7
MOEA/D-PBIL ↔ NSGA-II	~	~	~	+	+	~	+
MOEA/D-PBIL ↔ NSGA-II-Xing	~	~	~	+	+	~	+
MOEA/D-PBIL ↔ SPEA2	+	+	+	+	+	+	+
MOEA/D-PBIL ↔ MOPSO	+	+	+	+	+	+	+
MOEA/D-PBIL ↔ MOPBIL1	~	+	~	~	+	+	~
MOEA/D-PBIL ↔ MOPBIL2	+	+	~	~	+	+	+
Alg. 1 ↔ Alg. 2	Rnd-8	Rnd-9	Rnd-10	7copy	15copy	7hybrid	15hybrid
MOEA/D-PBIL ↔ NSGA-II	+	+	+	+	+	+	+
MOEA/D-PBIL ↔ NSGA-II-Xing	+	+	+	~	+	~	+
MOEA/D-PBIL ↔ SPEA2	+	+	+	+	+	+	+
MOEA/D-PBIL ↔ MOPSO	+	+	+	+	+	+	+
MOEA/D-PBIL ↔ MOPBIL1	+	+	+	+	+	+	+
MOEA/D-PBIL ↔ MOPBIL2	+	+	+	+	+	+	+

Table 9 Results of ACT (*Sec.*) of the seven algorithms (Best results are in bold).

Network	NSGA-II	NSGA-II-Xing	SPEA2	MOPSO	MOPBIL1	MOPBIL2	MOEA/D-PBIL
Rnd-1	15.31	14.31	9.26	26.83	9.32	9.35	**5.63**
Rnd-2	23.53	23.89	15.87	26.04	17.64	19.00	**7.05**
Rnd-3	61.52	57.88	34.21	27.41	37.03	36.77	**17.09**
Rnd-4	60.73	62.66	48.86	179.02	63.08	64.93	**34.23**
Rnd-5	93.42	89.41	77.75	228.61	79.17	85.95	**34.24**
Rnd-6	68.38	67.45	43.23	24.99	45.85	48.27	**17.83**
Rnd-7	196.29	193.27	221.68	503.51	148.49	146.72	**41.44**
Rnd-8	266.27	274.91	232.43	782.52	289.67	299.04	**89.21**
Rnd-9	747.23	748.76	671.05	1027.34	713.95	691.2	**134.92**
Rnd-10	958.91	1030.15	936.27	1324.86	1052.4	1060.1	**484.71**
7copy	93.28	94.29	87.56	182.10	92.01	93.86	**39.78**
15copy	921.24	884.73	929.93	1610.58	870.08	847.02	**710.95**
7hybrid	95.6	99.75	77.1	197.34	94.05	84.55	**37.16**
15hybrid	938.5	843.3	959.75	1559.75	846.9	858.3	**645.72**

Table 10 Description of the Mininet-based simulation test-bed.

Software/Environment	Description	Version
Mininet	Network emulator	2.2.1
Linux-Ubuntu	Operating system	16.04
Ryu	SDN controller	4.13
OpenFlow	Control/data plane interface	1.3
OpenvSwitch	multilayer virtual switch that supports OpenFlow	2.5.0
Python	Programming language	2.7.3