A Modified Ant Colony Optimization Algorithm for Network Coding Resource Minimization

Zhaoyuan Wang, Huanlai Xing, Tianrui Li, *Senior Member, IEEE*, Yan Yang, Rong Qu, *Senior Member, IEEE*, and Yi Pan, *Senior Member, IEEE*

***Abstract*—This paper presents a modified ant colony optimization (ACO) approach for the network coding resource minimization problem. It is featured with several attractive mechanisms specially devised for solving the concerned problem: 1) a multidimensional pheromone maintenance mechanism is put forward to address the issue of pheromone overlapping; 2) problem-specific heuristic information is employed to enhance the capability of heuristic search (neighboring area search); 3) a tabu-table-based path construction method is devised to facilitate the construction of feasible (link-disjoint) paths from the source to each receiver; 4) a local pheromone updating rule is developed to guide ants to construct appropriate promising paths; and 5) a solution reconstruction method is presented, with the aim of avoiding prematurity and improving the global search efficiency of proposed algorithm. Due to the way it works, the ACO can well exploit the global and local information of routing-related problems during the solution construction phase. The simulation results on benchmark instances demonstrate that with the integrated five extended mechanisms, our algorithm outperforms a number of existing algorithms with respect to the best solutions obtained and the computational time.**

***Index Terms*—Ant colony optimization (ACO), combinatorial optimization, network coding.**

I. Introduction

TRADITIONAL routing works in such a way that data information being transmitted is stored and forwarded at intermediate nodes in communication networks. At the network layer, data streams are processed separately like fluids share pipes or vehicles share highways [1]. Unfortunately, traditional routing cannot guarantee the achievement of maximum multicast throughput, determined by the max-flow min-cut theorem [2]. Hence, Ahlswede *et al*. [3] proposed network coding, an emerging communication paradigm that always enables the theoretical maximum data rate. Network coding has revolutionized the way of information processing and transmission in communications network. It is a great breakthrough in the field of information theory, computer science, and telecommunications.

The network coding resource minimization (NCRM) problem is a resource optimization problem emerging from the field of network coding. In the original studies, in order to achieve the theoretical maximum throughput of multicast, it was assumed that coding operations have to be performed at all coding-possible nodes [4]–[7]. This means all nodes which have the potential to perform coding would perform coding by default. However, as pointed out in [8]–[10], only a subset of coding-possible nodes suffices to realize network coding-based multicast (NCM) with an expected data rate. As network coding involves complicated mathematical operations (e.g., finite field computation), performing coding (and decoding) operations will consume significant computational and buffering resources in the corresponding nodes [11]. The less the coding operations, the less computational and buffering costs. When considering practical deployment, there is no doubt that carriers expect to make full use of the benefits the NCM brings while paying minimal computational and buffering costs. Therefore, it is worthwhile to study the problem of minimizing coding operations within NCM. Nowadays, evolutionary algorithms (EAs) are the mainstream solutions for NCRM in the field of computational intelligence (see Section III-B for details). However, the existing EAs for the NCRM problem are not good at integrating local information of the search space or domain-knowledge of the problem, which could seriously deteriorate their optimization performance.

Different from EAs, ant colony optimization (ACO) algorithms are the classes of reactive search optimization methods adopting the principle of “learning while optimizing” [12], [13]. They are constructive algorithms and simulate the behavior of the ant colony foraging for food and finding the most efficient routes from their nest to food sources. Since its introduction in 1992, a number of ACO variants have been proposed, e.g., ant colony system [14], MAX–MIN ant system [15], and best–worst ant system [16]. Meanwhile, ACOs have been intensively investigated and successfully applied to a vast number of optimization problems, e.g., vehicle routing problems [17], assignment problems [18], scheduling problems [19], and continuous optimization problem [20].

Manuscript received October 22, 2014; revised March 9, 2015 and May 8, 2015; accepted June 26, 2015. Date of publication July 17, 2015; date of current version May 26, 2016. This work was supported in part by the National Natural Science Foundation of China under Grant 61175047, Grant 61401374, and Grant 61170111; in part by the Fundamental Research Funds for the Central Universities of China under Grant 2682014RC23; in part by the project sponsored by Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry of China; and in part by the Stars of Sishi, Southwest Jiaotong University, and University of Nottingham. *(Corresponding author: Huanlai Xing.)*

Z. Wang, H. Xing, T. Li, and Y. Yang are with the School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China (e-mail: hxx@home.swjtu.edu.cn).

R. Qu is with the School of Computer Science, University of Nottingham, Nottingham NG8 1BB, U.K.

Y. Pan is with the Department of Computer Science, Georgia State University, Atlanta, GA 30303 USA.

Digital Object Identifier 10.1109/TEVC.2015.2457437

ACOs are capable of exploiting the local and global information of the underlining problems during the solution construction phase. This characteristic is especially suitable for addressing path-finding-related optimization problems, e.g., traveling salesman problem (TSP) and routing problems [14], [21]–[24]. Meanwhile, the objective of the NCRM problem is to find a subnetwork consisting of a set of link-disjoint paths. Therefore, ACOs may be a good candidate for solving the NCRM problem. However, to the best of our knowledge, there has not been any research conducted about applying ACO for the NCRM problem.

In this paper, a modified ACO is developed for tackling the NCRM problem. Based on the framework of the basic ACO, the proposed algorithm is devised with several attractive features specially for enhancing the optimization performance. These include a multidimensional pheromone maintenance mechanism, the use of problem-specific heuristic information, a tabu-table-based path construction method, a pheromone local updating rule, and a solution reconstruction method.

1) *Multidimensional Pheromone Maintenance Mechanism:* In the basic ACO, a single pheromone table is maintained. However, this always leads to a seriously deteriorated performance when solving the NCRM problem. Hence, we develop the abovementioned pheromone maintenance mechanism to effectively solve the pheromone overlapping problem.
2) *Problem-Specific Heuristic Information:* Due to the nature of the NCRM problem, there is no clear local heuristic information immediately available for ACO to solve the NCRM problem. Hence, we devise a heuristic information scheme to provide necessary guidance to an efficient search.
3) *Tabu-Table-Based Path Construction Method:* In the NCRM problem, a set of paths is expected to be built from the source to each receiver, which is extremely difficult. To deal with this issue, we propose a tabu-table-based path construction method to handle this constraint and support better collaborative performance of ants.
4) *Pheromone Local Updating Rule:* As constructing link-disjoint paths is quite difficult, the abovementioned path construction method may not be able to produce feasible solutions in some complicated circumstances. Hence, a pheromone local updating rule is introduced as a complement to the path construction method earlier. Inappropriate path selection is punished while promising path choices are rewarded to increase the probability of generating link-disjoint paths.
5) *Solution Reconstruction Method:* In order to avoid the search being stuck in local optima and diversify the solutions, we propose a solution reconstruction method to enhance local exploitation and alleviate the premature convergence.

The rest of this paper is organized as follows. Section II introduces the basic ACO algorithm framework and the graph decomposed method for the NCRM problem. Section III describes the problem formulation and related works. Details of the proposed algorithm are introduced in Section IV. Simulation results are analyzed in Section V. The conclusion is presented in Section VI.

II. Basic Concepts

In this section, we briefly review the framework of the basic ACO and the graph decomposition method for the NCRM problem.

A. ACO

ACO was originally created to address the TSP. Hence, this section describes the procedure of the basic ACO for TSP as an example [14], [21].

Given a number of cities, the objective of TSP is to find a minimal travel distance while traversing each city once. Assume that there are n cities fully connected by edge set E. The search procedure is shown below.

1) *Initialization:* Randomly select m cities and place each city with an ant. Set the initial pheromone value on each edge to a very small positive variable τ_0.
2) *Path Construction:* Ant k ($k = 1, 2, \ldots, m$) (in city i) decides the next city j to visit, according to the transition probability given in

$$p(i,j) = \begin{cases} \dfrac{[\tau(i,j)]^{\alpha}[\eta(i,j)]^{\beta}}{\sum_{u\in\Psi_i}[\tau(i,u)]^{\alpha}[\eta(i,u)]^{\beta}}, & j \in \Psi_i \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Let $\tau(i,j)$ represents the pheromone on edge(i,j) and $\eta(i,j) = 1/d_{ij}$ be the heuristic information on edge(i,j) reflecting local information, where d_{ij} is the distance from city i to j. Let Ψ_i denotes an edge set that records all edges an ant could visit. Let α and β denote the weight factors, which measure the relative importance between the pheromone and the heuristic information.
3) Implement local search to optimize the solution found by ant k (optional) [22]. If all ants have completed step 2), go to step 4). Otherwise, go to step 2).
4) Update the pheromone level by using

$$\tau(i,j) = (1-\rho)\tau(i,j) + \rho\Delta\tau(i,j) \quad (2)$$

where the parameter $\rho \in (0, 1)$ represents the evaporation coefficient. The term $\Delta\tau(i,j)$ is associated with the performance of each ant.
5) If the termination condition is met, stop the procedure and output the best solution obtained.

B. Graph Decomposition Method

A communication network can be modeled as a directed graph $G(V, E)$ where V and E denote the set of nodes and links, respectively. Assume each link $e \in E$ is with a unit capacity. We refer to each nonreceiver node with multiple incoming links as a merging node which can perform coding operation if necessary. However, it is difficult to determine whether coding is needed at a merging node and how coding is performed when needed. In order to clearly show all possibilities when an information flow joins a merging node, the graph decomposition method was proposed to decompose a merging node into a set of auxiliary nodes connected with auxiliary links [9], [10]. The following describes the graph decomposition procedure.

```
for $t = 1$ to $|V|$ do
    if $v^t$ is a merging node then
        for $i = 1$ to $n_{in}$ do
            Create a new incoming auxiliary node, denoted
                by $v^t_{in}(i)$, then add to $G$;
            Redirect link $e_{in}(i)$ to $v^t_{in}(i)$;
        end for
        for $j = 1$ to $n_{out}$ do
            Create a new outgoing auxiliary node, denoted
                by $v^t_{out}(j)$, then add to $G$;
            Redirect link $e_{out}(j)$ to $v^t_{out}(j)$;
        end for
        for $i = 1$ to $n_{in}$ do
            for $j = 1$ to $n_{out}$ do
                Create a new auxiliary link from $v^t_{in}(i)$ to
                    $v^t_{out}(j)$ and then add to $G$;
            end for
        end for
        Remove $v^t$ from $G$;
    end if
end for
```

Fig. 1. Pseudocode of the graph decomposition method.

Each merging node *m* is decomposed into two auxiliary node sets, i.e., the incoming auxiliary node set *In*(*m*) and the outgoing auxiliary node set *Out*(*m*). Let I_m and O_m be the incoming and outgoing link sets of merging node *m*, respectively. Then, *In*(*m*) has $|I_m|$ incoming auxiliary nodes while *Out*(*m*) owns $|O_m|$ outgoing auxiliary nodes. In *In*(*m*), each node corresponds to a unique link in I_m. Likewise, each node in *Out*(*m*) corresponds to a unique link in O_m. During the graph decomposition, each link in I_m is redirected to the corresponding incoming auxiliary node and each link in O_m is redirected to the corresponding outgoing auxiliary node. In addition, auxiliary links are inserted between incoming and outgoing auxiliary nodes so that any incoming auxiliary node is connected to all outgoing auxiliary nodes. Let $G_D(V', E')$ be the decomposed graph of $G(V, E)$. Fig. 1 shows the pseudocode of the graph decomposition method, where $v^t \in V$, $|V|$ is the number of nodes in *V*, links $e_{in}(i)$ and $e_{out}(j)$ denote the *i*th incoming link and the *j*th outgoing link of v^t, respectively, and n_{in} and n_{out} are the numbers of incoming and outgoing links of v^t, respectively.

Fig. 2 illustrates an example of the graph decomposition method. The original graph with a source (i.e., node 1) and two receivers (i.e., nodes 8 and 9) are shown in Fig. 2(a), where nodes 4 and 7 are merging nodes. Fig. 2(b) shows the decomposed graph, where eight auxiliary links are inserted. Node 4 is decomposed into two incoming auxiliary nodes, nodes 4_*i*_1 and 4_*i*_2, and two outgoing auxiliary nodes, nodes 4_*o*_1 and 4_*o*_2. Likewise, node 7 is decomposed into four auxiliary nodes, as shown in Fig. 2(b). The decomposed graph unveils all possibilities that information flows may pass through nodes 4 and 7.

Note that each outgoing auxiliary node in $G_D(V', E')$ has a single outgoing link. Therefore, if more than one information flow joins an outgoing auxiliary node, it means the coding operation is required at that auxiliary node. In addition, the graph decomposition method only decomposes merging nodes which does not affect the source, receivers, and data rate of the graph.

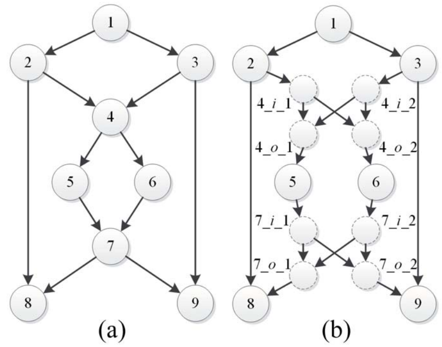


Fig. 2. Example of the graph decomposition method. (a) Original Graph. (b) Decomposed Graph.

III. Problem Formulation and Related Works

A. Problem Formulation

As aforementioned, a communication network is represented by a directed graph $G(V, E)$. After the graph decomposition, $G(V, E)$ is transformed to graph $G_D(V', E')$. A single-source NCM scenario can be defined as a four-tuple set (G_D, s, T, R), where the information needs to be transmitted at data rate *R* from the source node $s \in V'$ to a set of *d* receivers $T = \{t_1, t_2, \ldots, t_d\}$. We assume each link has a unit capacity, so a path from *s* to t_k has a unit capacity. If *R* link-disjoint paths $\{p_1(s, t_k), \ldots, p_R(s, t_k)\}$ from *s* to each receiver $t_k \in T$ are set up, the data rate *R* is said to be achievable. The *R* link-disjoint path set $\{p_1(s, t_k), \ldots, p_R(s, t_k)\}$ is denoted by $Paths(s, t_k)$, where $t_k \in T$. If we successfully obtained $Paths(s, t_1), \ldots, Paths(s, t_d)$, then we obtain a feasible solution $Solution(G_D)$. According to the solution $Solution(G_D)$, a NCM subgraph can be built to support the multicast with network coding, which is denoted by $G_{\text{NCM}}[Solution(G_D)]$.

The following lists some notations used in this paper.

1) *s:* The source node in $G_D(V', E')$.
2) $T = \{t_1, t_2, \ldots, t_d\}$*:* Set of receivers, where $d = |T|$ is the number of receivers.
3) *R:* Data rate (an integer) at which *s* expects to transmit to *T*.
4) $p_i(s, t_k)$*:* The *i*th path from *s* to t_k, where $t_k \in T$ and $i = 1, \ldots, R$.
5) $W_i(s, t_k)$*:* The set of links of $p_i(s, t_k)$, i.e., $W_i(s, t_k) = \{e | e \in p_i(s, t_k)\}$.
6) $Paths(s, t_k) = \{p_1(s, t_k), \ldots, p_R(s, t_k)\}$*:* A path set from *s* to t_k, where $t_k \in T$ and any two paths in $Paths(s, t_k)$ are link-disjoint.
7) $Solution(G_D) = \{Paths(s, t_1), \ldots, Paths(s, t_d)\}$*:* A complete NCM solution.
8) $G_{\text{NCM}}[Solution(G_D)]$*:* A NCM subgraph that is built by $Solution(G_D)$.

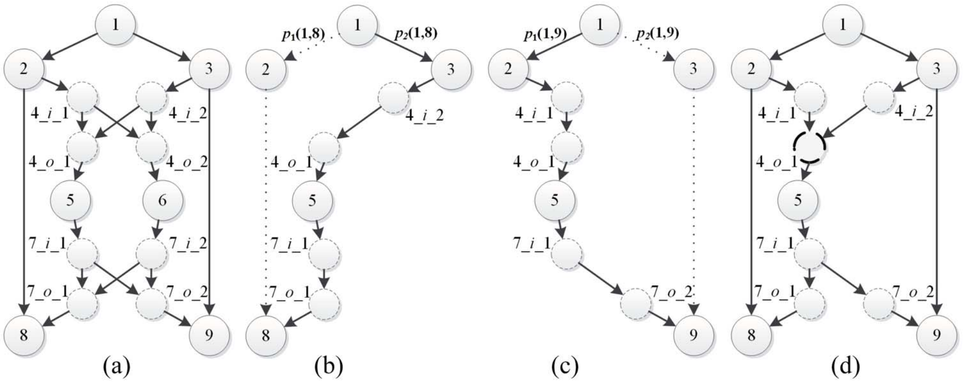


Fig. 3. Illustrative example of the problem formulation. (a) Decomposed Graph. (b) $Paths(1, 8) = \{p_1(1, 8), p_2(1, 8)\}$. (c) $Paths(1, 9) = \{p_1(1, 9), p_2(1, 9)\}$. (d) $Solution(G_D) = \{Paths(1, 8), Paths(1, 9)\}$.

9) *$OA(G_D)$:* The set of outgoing auxiliary nodes in $G_D(V', E')$.
10) *σ_o:* A binary variable associated with each node $o \in OA(G_D)$. $\sigma_o = 1$ if at least two incoming links of node o are occupied by $G_{\text{NCM}}[Solution(G_D)]$; $\sigma_o = 0$, otherwise.
11) *$\varphi[G_{\text{NCM}}(Solution(G_D))]$:* The number of coding nodes in $G_{\text{NCM}}(Solution(G_D))$.

The NCRM problem is defined as finding a solution to build a NCM subgraph $G_{\text{NCM}}[Solution(G_D)]$ with the minimum amount of coding operations performed and the data rate R satisfied, as

$$\text{Minimize:} \quad \varphi[G_{\text{NCM}}(Solution(G_D))] = \sum_{\forall o \in OA(G_D)} \sigma_o \tag{3}$$

$$\text{Subject to:} \quad R(s, t_k) = R, \ \forall t_k \in T \tag{4}$$

$$W_i(s, t_k) \cap W_j(s, t_k) = \emptyset, \quad \forall t_k \in T, \ \forall i, j \in \{1, \dots, R\}, \ i \neq j. \tag{5}$$

Objective (3) defines the optimization problem as minimizing the number of coding operations. Constraint (4) defines that the achievable rate between s and each receiver is exactly data rate R in solution $Solution(G_D)$, indicating that there are R paths between the source and each receiver. Constraint (5) indicates that for two arbitrary paths from s to t_k, $p_i(s, t_k)$, and $p_j(s, t_k)$ $(i \neq j)$, no common link exists so that each receiver can receive information at data rate R.

An illustrative example is given in Fig. 3. Fig. 3(a) illustrates the decomposed graph for the original multicast scenario in Fig. 2. With data rate $R = 2$ and two receivers, i.e., nodes 8 and 9, we use an ant colony of two ant groups ($AntG_1$ and $AntG_2$) to address the NCRM problem, where each group consists of two ants. $AntG_1$ is responsible for finding a path set of two link-disjoint paths from nodes 1 to 8. $AntG_2$ is for constructing a link-disjoint path set from nodes 1 to 9. Specifically, as shown in Fig. 3(b) and (c), the two ants in $AntG_1$ find $p_1(1, 8) = 1 \to 2 \to 8$ and $p_2(1, 8) = 1 \to 3 \to 4\_i\_2 \to 4\_o\_1 \to 5 \to 7\_i\_1 \to 7\_o\_1 \to 8$, respectively. Thus $W_1(1, 8) = \{1 \to 2,\ 2 \to 8\}$ and $W_2(1, 8) = \{1 \to 3,\ 3 \to 4\_i\_2,\ 4\_i\_2 \to 4\_o\_1,\ 4\_o\_1 \to 7\_i\_1,\ 7\_i\_1 \to 7\_o\_1,\ 7\_o\_1 \to 8\}$. Due to $W_1(1, 8) \cap W_2(1, 8) = \emptyset$, the two paths $p_1(1, 8)$ and $p_2(1, 8)$ are link-disjoint. Likewise, then the other ants in $AntG_2$ find two link-disjoint paths $p_1(1, 9) = 1 \to 2 \to 4\_i\_1 \to 4\_o\_1 \to 5 \to 7\_i\_1 \to 7\_o\_2 \to 9$ and $p_2(1, 9) = 1 \to 3 \to 9$, respectively. Eventually, a complete solution $Solution(G_D) = \{Paths(1, 8), Paths(1, 9)\}$ can be constructed, where $Paths(1, 8) = \{p_1(1, 8), p_2(1, 8)\}$ and $Paths(1, 9) = \{p_1(1, 9), p_2(1, 9)\}$, then the associated NCM subgraph is built as shown in Fig. 3(d). It is noted that node 4_o_1 is the only coding node in $G_{\text{NCM}}[Solution(G_D)]$, which means the number of the coding nodes φ equals 1.

B. Related Works

Due to the importance and the benefit network coding brings, the NCRM problem has received much attention recently. Langberg *et al.* [11] and Fragouli and Soljanin [25] proposed two greedy approaches for solving the problem. However, greedy algorithms do not perform well in escaping local optimum, leading to a deteriorated optimization performance when the link traversing order is not appropriate. Later on, Kim *et al.* [8]–[10] proved that the NCRM problem is NP-hard and carried out a series of research on how to efficiently apply genetic algorithms (GAs) to tackle the problem. Simulation results demonstrate that GAs outperform the greedy algorithms in a statistical manner. Since then, EA-based search algorithms have become the mainstream techniques for solving the NCRM problem in the field of computational intelligence.

We classify the existing EAs into four categories by the individual encoding approaches adopted. EAs of the first category are based on the binary link state (BLS) encoding. As mentioned in Section II-B, for a merging node m, there are $|I_m| \times |O_m|$ auxiliary links inserted between the corresponding incoming and outgoing auxiliary nodes. In BLS encoding, an individual consists of a number of binary variables, with each corresponding to the state of an auxiliary link (active or inactive). Hence, an explicit NCM subgraph can be built by a feasible individual. The BLS-based EAs include GAs [9], [10], [26], quantum-inspired EAs [27], [28], population-based incremental learning [29], [30], and compact GA [31]. One of the disadvantages of BLS is that infeasible solutions account for the majority of the search space, which to a certain extent deteriorates the search ability and efficiency of EAs [32], [33].

EAs of the second category are based on the block transmission state (BTS) encoding. BTS is similar to BLS. In BTS, an individual is divided into a number of blocks, each of which corresponds to an outgoing auxiliary node. If there are at least two 1s in a block, the whole block is set to all-one block. In this way, the size of the search space is greatly decreased. Nevertheless, using BTS may lose useful information for guiding the search toward the global optima. GA [10] is based on BTS encoding. In addition, Ahn and Kim [34] and Ahn [35] incorporated the self-adaptive fitness assignment rule and entropy-based relaxation technique into EAs with BTS to improve the efficiency and effectiveness of the algorithms.

As mentioned earlier, BLS and BTS encodings both record the explicit link states (active or inactive). But, the third category of the EAs utilizes the relative information of the flows [36]. To be specific, each link is associated with a coefficient which represents how the information is combined according to the combination of flows from the upstream links. Hu *et al*. [37] and Hu and Leeson [38] invented this encoding approach and adapted several GAs, e.g., the ripple-spreading GA and the spatial receding horizon control GA (SRHCGA), for the problem in large-scale or complex networks. Meanwhile, a chemical reaction optimization (CRO) algorithm was studied for addressing the problem, with the operating principle inspired from chemical reactions [39]. Different from optimizing routing only, their research also worked out the associated information encoding/decoding scheme, which is an important and realistic issue when considering the practical deployment of NC.

The fourth stream of EAs is the path-oriented encoding method. Each individual is composed of a union of paths from the source to one of the receivers. Compared with BLS and BTS, the path-oriented encoding results into a search space where all solutions are feasible. As there is no infeasible solution, the search space is well connected and the problem difficulty is reduced. Xing *et al.* [33] proposed a path-oriented encoding EA.

In addition to the NCRM problem mentioned earlier, more and more research efforts have been made to the multiobjective NCM routing problem (MNCMRP), where coding cost, link cost, and quality-of-service indicators are often considered as multiple objectives for simultaneous optimization. Coding cost and link cost are often considered as two conflicting objectives in the context of MNCMRP. A number of multiobjective EAs have been proposed to gain the tradeoff between the two costs [40]–[42]. Xing and Qu [32] formulated a novel MNCMRP, where the total cost and maximum end-to-end delay are two objectives. The fast nondominated sorting genetic algorithm II was adapted for the problem. Moreover, Karunarathne *et al*. [43] investigated a MNCMRP with three objectives, including the number of coding nodes, the mean number of coding node input links, and the sharing of resources by receivers.

IV. NCRM-ACO

In this section, we first describe the overall procedure of the ACO algorithm for the NCRM problem (NCRM-ACO), followed by details of the key mechanisms and significance of parameters in sections.

A. Overall Procedure of NCRM-ACO

Fig. 4 is the overall procedure of NCRM-ACO and Fig. 5 shows the pseudocode of function PathSetConstruction. Fig. 6 shows the overall flow chart of the algorithm. In the proposed NCRM-ACO, first of all, with the original network $G(V, E)$, the graph decomposition phase is executed so as to obtain a decomposed graph $G_D(V', E')$, based on which ACO is implemented to build feasible solutions. The proposed algorithm maintains a single ant colony at each generation. Within the colony, there are d ant groups $AntG_k$, $k = 1, \ldots, d$, each of which contains R ants (R is the expected data rate). Each ant group corresponds to one of d receivers, i.e., the kth ant group is in charge of finding a feasible path set $Paths(s, t_k)$ for receiver $t_k \in T$, where $Paths(s, t_k)$ is composed of R link-disjoint paths from the source to t_k. Each ant in $AntG_k$ finds a single path from the source to t_k so that the abovementioned R link-disjoint paths are constructed for receiver t_k. In the algorithm, d path sets are built one after another. If path set $Paths(s, t_k)$ is constructed successfully (see Section IV-D), it is used to update the pheromone and heuristic information of the ant colony to guide the path construction process (see Section IV-E). With all path sets found, a complete solution $Solution_z(G_D)$ consisting of all paths in these path sets is formed, where z is the generation number. Then, a NCM subgraph could be built by the solution and the number of coding nodes $\varphi[G_{\text{NCM}}(Solution_z(G_D))]$ is easily calculated. After that, a solution reconstruction method is devised to improve the quality of $Solution_z(G_D)$ by exploring its neighboring area in the solution space, aiming to find an improved solution $Solution_z^{\text{new}}(G_D)$ (see Section IV-F). Finally, the global (historical) best solution $Solution_{gb}(G_D)$ obtained is used to update the pheromone so as to guide the search toward the optimal solution to the problem (see Section IV-G). The abovementioned process is repeated generation by generation, until the termination condition is met.

The pheromone and heuristic coefficients are two important coefficients, necessarily supporting effective search. In Sections IV-B and IV-C, two problem-specific pheromone and heuristic maintenance mechanisms are described in detail. The remaining steps of NCRM-ACO are introduced from Sections IV-D–IV-G.

B. Pheromone Maintenance Mechanism

In this paper, pheromone is used to provide essential guidance for the ant colony to gradually search toward the optimal solution for the NCRM problem. As mentioned in Section III-A, the less coding operations are required the better. Hence, pheromone is designed to be associated with the number of coding nodes a solution owns. This idea is similar to the pheromone scheme in TSP and 0-1 knapsack problems [14], [44], where pheromone is associated with the total distance and the total number of bins, respectively.

However, there is a significant difference between the pheromone schemes for TSP and 0-1 knapsack problems

```
Input: A graph $G$, data rate $R$
Decompose graph $G$ to $G_D$;                                                    ▷ (Subsection II-B )
Initialize pheromone values;                                                     ▷ (Subsection IV-B)
Initialize $Solution_{gb}(G_D) = \emptyset$;
while Termination conditions NOT met do
    Initialize $Solution_z(G_D) = \emptyset$;
    Initialize heuristic information table;                                      ▷ (Subsection IV-C)
    for $k = 1$ to $d$ do
        Initialize $Paths(s, t_k) = \emptyset$;
        Set $Paths(s, t_k)$ = PathSetConstruction($s, t_k, R$);                  ▷ (Subsection IV-D)
        while size of $Paths(s, t_k) < R$ do
            Invoke the pheromone local updating rule (punishment) to $Paths(s, t_k)$;   ▷ (Subsection IV-E)
            Set $Paths(s, t_k)$ = PathSetConstruction($s, t_k, R$);              ▷ (Subsection IV-D)
        end while
        Invoke the pheromone local updating rule (reward) to $Paths(s, t_k)$;    ▷ (Subsection IV-E)
        Add $Paths(s, t_k)$ into $Solution_z(G_D)$;
        Update the heuristic information according to $Paths(s, t_k)$;           ▷ (Subsection IV-C)
    end for
    Apply solution reconstruction method to $Solution_z(G_D)$ and get $Solution_z^{new}(G_D)$;   ▷ (Subsection IV-F)
    if $\varphi(G_{\mathrm{NCM}}(Solution_z^{new}(G_D))) < \varphi(G_{\mathrm{NCM}}(Solution_{gb}(G_D)))$ then
        Invoke pheromone global updating rule by $\varphi(G_{\mathrm{NCM}}(Solution_z^{new}(G_D)))$;   ▷ (Subsection IV-G)
        Set $Solution_{gb}(G_D) = Solution_z^{new}(G_D)$;
    end if
end while
Output: The global best solution $Solution_{gb}(G_D)$ and $\varphi(G_{\mathrm{NCM}}(Solution_{gb}(G_D)))$
```

Fig. 4. Overall procedure of NCRM-ACO.

```
function PATHSETCONSTRUCTION($source, receiver, R$)
    Initialize $Paths(source, receiver) = \emptyset$;
    for $l = 1$ to $R$ do
        Ant $l$ builds a path from $source$ to $receiver$, denoted by $p_l(source, receiver)$;   ▷ (Subsection IV-D)
        Add $p_l(source, receiver)$ into $Paths(source, receiver)$;
    end for
    return $Paths(source, receiver)$;
end function
```

Fig. 5. Pseudocode of constructing the path set from the source to a receiver.

than that for the NCRM problem. That is, for the former, a single pheromone table is able to provide effective guidance during the search while for the latter such a scheme does not apply, as explained below. Compared with TSP and 0-1 knapsack problems, NCRM problem is much more complicated. In TSP, each link is selected once in an arbitrary solution. Nevertheless, in the NCRM problem, each link could be idle, occupied once or multiple times (e.g., a link may belong to multiple path sets simultaneously). If a single pheromone table is adopted, this conflicting and misleading information (pheromone overlapping problem) would not be able to provide useful guidance for the solution construction procedure. This is because for an arbitrary link different ants may have different options on whether or not to occupy it.

In order to efficiently guide the search, NCRM-ACO uses a new pheromone maintenance mechanism employing multiple pheromone tables. We associate each ant in the ant colony with a pheromone table, leading to in total $R * d$ pheromone tables, where R and d are the data rate and the number of receivers, respectively. Each table maintains the pheromone of an ant over the decomposed graph G_D, where each auxiliary link is associated with a pheromone value. Let τ_0 be the initial pheromone value over each link. For all tables, τ_0 is set to a small positive number $\varphi_{\max} = (|V'|)^{-1}$, where $|V'|$ is the number of nodes in G_D. Take Fig. 2(a) as an example, with $d = 2$ and $R = 2$, the NCRM-ACO maintains $2 \times 2 = 4$ pheromone tables as shown in Fig. 7. At different generations, those ants responsible for finding the same path, e.g., $p_1(1, 8)$, share the same pheromone table. Moreover, τ_0 for all links is set to $\varphi_{\max} = (|V'|)^{-1} = (15)^{-1}$. During the search procedure, the pheromone values in those tables are gradually updated, as introduced in Sections IV-E and IV-G. The number of pheromone tables is the product of the number of the data rate R and the number of receivers d. Data rate R is subjected by the max-flow from the source to a receiver. In the literature, data rate R is usually small. To the best of our knowledge, the largest R for experiments and simulations is set to 7 [35]. So, the number of pheromone tables grows approximately linearly with d.

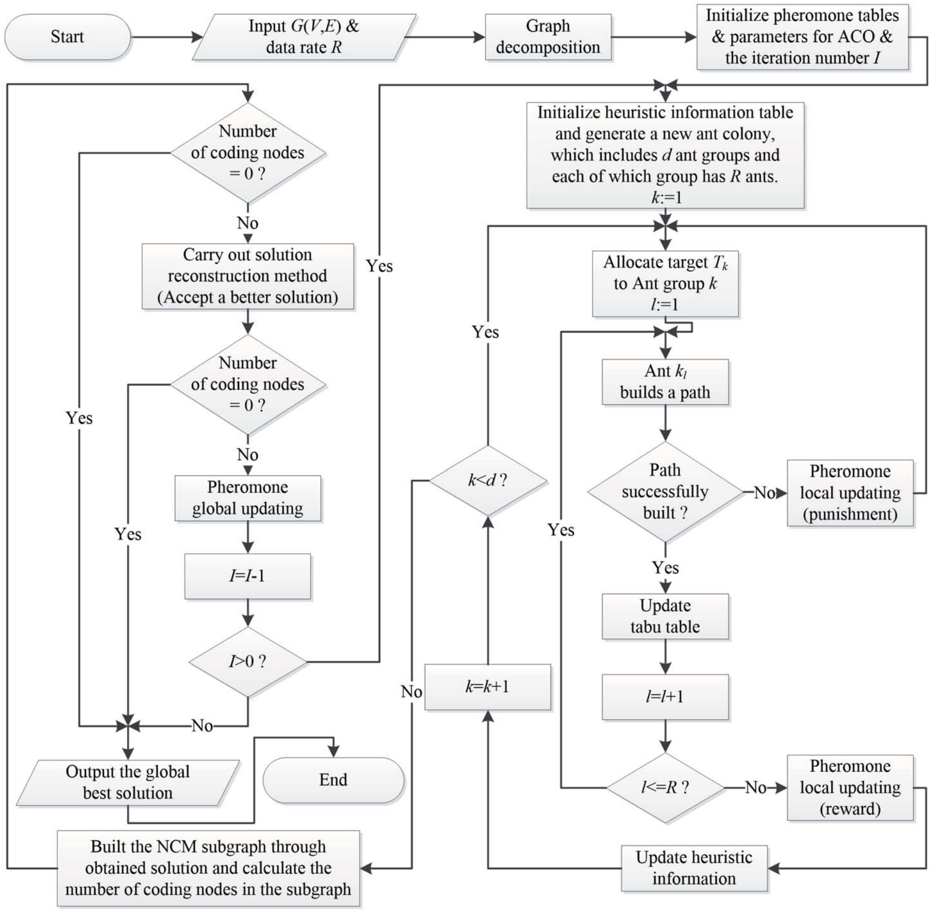

Start
Input $G(V,E)$ & data rate R
Graph decomposition
Initialize pheromone tables & parameters for ACO & the iteration number I
Initialize heuristic information table and generate a new ant colony, which includes d ant groups and each of which group has R ants. k:=1
Allocate target T_k to Ant group k l:=1
Ant k_l builds a path
Path successfully built ?
No
Pheromone local updating (punishment)
Yes
Update tabu table
l=l+1
l<=R ?
No
Pheromone local updating (reward)
Yes
Update heuristic information
k=k+1
k<d ?
Yes
No
Built the NCM subgraph through obtained solution and calculate the number of coding nodes in the subgraph
Number of coding nodes = 0 ?
No
Carry out solution reconstruction method (Accept a better solution)
Number of coding nodes = 0 ?
No
Pheromone global updating
I=I-1
I>0 ?
Yes
No
Yes
Yes
Output the global best solution
End

Fig. 6. Overall flow chat of NCRM-ACO.

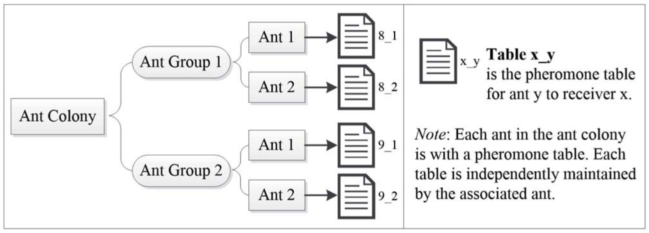

Ant Colony
Ant Group 1
Ant Group 2
Ant 1
Ant 2
Ant 1
Ant 2
8_1
8_2
9_1
9_2
x_y
Table x_y is the pheromone table for ant y to receiver x.
Note: Each ant in the ant colony is with a pheromone table. Each table is independently maintained by the associated ant.

Fig. 7. Example of multiple pheromone tables maintained.

C. Heuristic Maintenance Mechanism

In ACOs, the heuristic information is of vital importance for guiding the construction of the global best solutions, e.g., the distance between two cities in TSP and the weight and value of goods in the 0-1 knapsack problem [14], [44]. In those problems, such information can be easily extracted and defined to better explore the neighboring areas. However, the NCRM problem aims to find a feasible routing subgraph consisting of multiple path sets, each of which contains a number of disjoint paths to the same receiver, where no clear heuristic information is immediately available.

In this paper, an efficient heuristic maintenance mechanism maintains how many times each link has been selected by different ant groups in the same generation. Heuristic information represents local information and can provide some useful guidance when constructing the paths to form the NCM subgraph. According to Section II-B, an outgoing auxiliary node $m \in OA(G_D)$ will perform coding operations if the received information comes from more than one incoming link. How to reduce the probability of the incurrence of coding operations is desirable. Fortunately, the number of times that each incoming link is selected can help. This is because, at a certain generation, each ant aims at finding a complete path from the source to one of the receivers and for a certain outgoing auxiliary node, if all ants pass (select) a single incoming link, then no coding operation is necessary at the outgoing auxiliary node. If we use the number of times that each incoming link is selected by ants, it is possible to reduce the coding probability during the NCM data transmission. That is, an incoming link associated with larger number of times is selected by ants. In this way, for an arbitrary outgoing auxiliary node, one of its incoming links is selected multiple times while the rest of the links are not employed by the NCM. Hence, the coding operation is avoided at the outgoing auxiliary node. On the other hand, if ants are allowed to randomly select incoming links, coding operations are more likely to happen. To realize such an idea in the proposed ACO, we use the number of times that each incoming link is selected as the heuristic information, to provide necessary guidance for the ant groups to find all feasible path sets while trying best to involve as less coding operations as possible. Specifically, for an arbitrary outgoing auxiliary node, we count the number of times each incoming link is selected by all ant groups at each generation. When implementing path construction, each ant

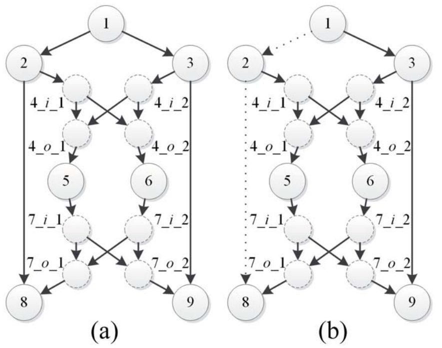


Fig. 8. Example of the tabu table. (a) Decomposed Graph. (b) Tabu table = $\{1 \to 2, 2 \to 8\}$.

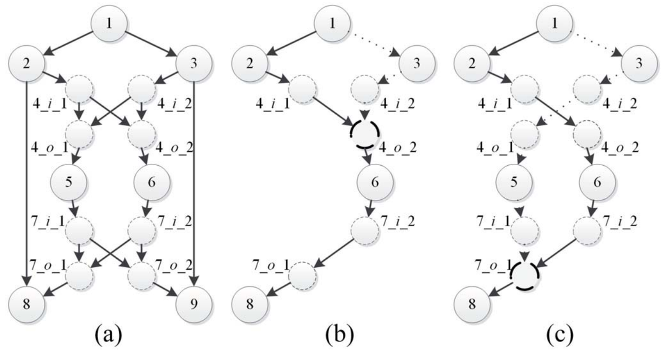


Fig. 9. Example of the inappropriate path selection. (a) Decomposed Graph. (b) Circumstance 1. (c) Circumstance 2.

preferably selects those incoming links with higher heuristic information values. Hence, in a greedy manner, the role of the heuristic information is to provide extra guidance to reduce the number of coding operations in the solution construction phase.

In the proposed mechanism, the heuristic information is maintained in one table, called key-value map, where key and value represent the link ID and its corresponding value, respectively. The value stands for the number of times a link has been selected. Initially, the value of each link is set to 1. All path sets are constructed in a one-by-one manner. The table is updated after each of the d path sets is constructed by adding a value of 1 to the heuristic information value of each link in the path set. At the beginning of each new generation, the values of all links are reset to 1 since the heuristic information is only used to indicate the link occupation status of the incumbent generation.

D. Tabu-Table-Based Path Construction

Different from the TSP, the NCRM problem is much more complex. It aims to construct multiple path sets, each consisting of a number of link-disjoint paths from the source to a certain receiver. Due to the problem nature, it is often possible that an ant could not reach its destination, e.g., receiver t_k. To overcome this problem, we propose a tabu-table-based path construction method to increase the probability that an arbitrary ant can find a feasible and demanded path.

In the proposed method, the route of each ant starts from the source and ends up with one of the receivers. A feasible solution to the NCRM problem is quite difficult to construct, since one needs to find multiple path sets, where each path set contains multiple link-disjoint paths from the source to the same receiver. To ease the abovementioned problem, for each ant group $AntG_k$, we maintain a tabu table to record which links have been employed. Those employed links will not be visited by other ants within $AntG_k$. Fig. 8 illustrates a simple example of the tabu table. When ant_1 in $AntG_1$ find $p_1(1,8) = 1 \to 2 \to 8$ as shown in Fig. 8(b), the two links $1 \to 2$ and $2 \to 8$ are added in the tabu table. Then, ant_2 of $AntG_1$ would not choose the two links any more. If there is only a single link from node i, the ant will move to this link; otherwise, those available links Ψ_i which are not being included in the tabu table will have a chance to be selected. To select a link from Ψ_i, the pseudorandom rule [14] is adopted to calculate the probability by

$$np = \begin{cases} \arg\max_{u \in \Psi_i} [\tau(t_k, l, (i,u))]^{\alpha} [\eta(i,u)]^{\beta}, & \text{if } q \le q_0 \\ \zeta, & \text{otherwise} \end{cases} \quad (6)$$

where argument $\tau[t_k, l, (i,u)]$ is amount of pheromone and $\eta(i,u)$ is the amount of heuristic information on link (i,u). In τ, receiver t_k and path number l are both associated with the pheromone maintenance mechanism. Parameters α and β define the relative importance of the pheromone and the heuristic information, respectively. q is a uniformly distributed random number in the range $[0,1]$ and $q_0 (0 < q_0 < 1)$ is a threshold value. ζ is a random value determined by the probability of $p(i,j)$ if q is greater than q_0

$$p(i,j) = \begin{cases} \dfrac{\left[\tau(t_k, l, (i,j))\right]^{\alpha} \left[\eta(i,j)\right]^{\beta}}{\sum_{u \in \Psi_i} [\tau(t_k, l, (i,u))]^{\alpha} [\eta(i,u)]^{\beta}}, & j \in \Psi_i \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

By using formulas (6) and (7), each ant may either follow the most favorite path already established or randomly select a path based on the probability distribution of the pheromone and the heuristic accumulated. It is noted that the pseudorandom rule facilitates the diversity of the stochastic search and hence it helps to enhance the global search ability.

E. Pheromone Local Updating Rule

The expected data rate R, as a hard constraint, must be satisfied during the establishment of the NCM session. This can be achieved by constructing R link-disjoint paths from the source to each receiver. However, even if the tabu table scheme is employed, an infeasible path could be resulted if an ant chooses inappropriate links. Take Fig. 9(a) as an example. An ant in ant group $AntG_1$ has constructed a path $p_1(1,8) = 1 \to 2 \to 4\_i\_1 \to 4\_o\_1 \to 5 \to 7\_i\_1 \to 7\_o\_1 \to 8$ from source node 1 to receiver node 8 and all links in $p_1(1,8)$ are recorded in the tabu table. The other ant in $AntG_1$ cannot construct a second path $p_2(1,8)$ that is link-disjoint with $p_1(1,8)$ from nodes 1 to 8 in any circumstance, as shown in Fig. 9(b) and (c). Apparently, if this happens, we could send a new group of ants to reconstruct a feasible path set. In NCRM-ACO, a pheromone punishing-and-rewarding mechanism is proposed to avoid ants following the same paths as that in the old group.

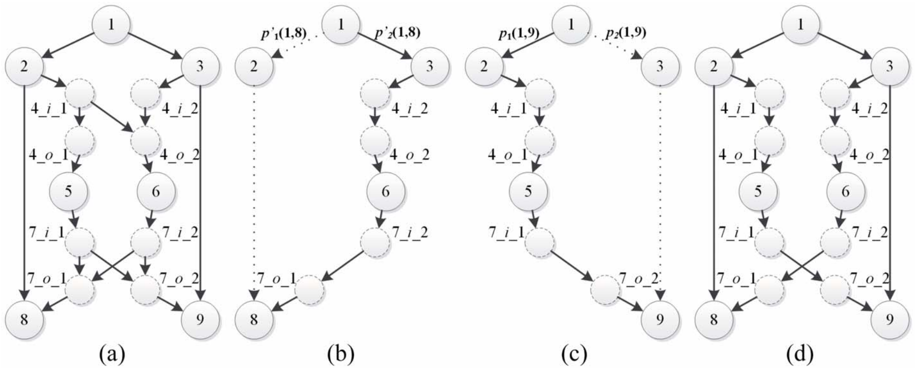


Fig. 10. Example of the solution reconstruction method. (a) Reconstructed Graph G'_D. (b) $Paths_{\text{new}}(1, 8) = \{p'_1(1, 8), p'_2(1, 8)\}$. (c) Remained $Paths(1, 9)$. (d) $Solution^{\text{new}}(G_D) = \{Paths_{\text{new}}(1, 8), Paths(1, 9)\}$.

In the pheromone punishing scheme, if an ant group $AntG_k$ fails to construct a feasible path set, e.g., $Paths(s, t_k)$, the pheromone values on those paths which have been employed by $AntG_k$ are decreased by a constant $\Delta\tau_{loc}$ before the reconstruction of $Paths(s, t_k)$ as

$$\tau(t_k, l, (i, j)) = \tau(t_k, l, (i, j)) - \Delta\tau_{loc} \tag{8}$$

where the value $\Delta\tau_{loc}$ is a small positive number. In the pheromone local updating rule, $\Delta\tau_{loc} = (\varphi_{\max})^{-1}$.

It is noted that the pheromone values on some of the links may decrease constantly, which can cause a stagnation in the search when the difference of pheromone on links is too large. Therefore, inspired by the idea of MAX–MIN ant system [15], in our scheme the pheromone value on any link cannot be lower than a threshold value $(\varphi_{\max})^{-1}$, i.e., whenever $\tau[t_k, l, (i, u)] - \Delta\tau_{loc} \leq (\varphi_{\max})^{-1}$, set $\tau[t_k, l, (i, u)] = (\varphi_{\max})^{-1}$, which could effectively avoid the stagnation in the search.

On the contrary, in the pheromone rewarding scheme, when a feasible path set is constructed successfully, the associated ant group will be rewarded by means of increasing the pheromone values on links they employ by $\Delta\tau_{loc}$ at each time

$$\tau(t_k, l, (i, j)) = \tau(t_k, l, (i, j)) + \Delta\tau_{loc}. \tag{9}$$

In summary, the pheromone local updating rule is composed of the punishing and rewarding schemes to guide the construction of feasible solutions.

F. Solution Reconstruction Method

It is widely recognized that prematurity often happens in ACO and could cause serious performance deterioration [15], [45]. Hence, we develop a solution reconstruction method to improve the quality of the solution obtained, aiming at enhancing the local exploitation ability and avoiding the premature convergence. The solution reconstruction method consists of three steps. First of all, for a given solution $Solution(G_D)$, we randomly select a coding node m_{coding} from it. Second, we randomly select one of the incoming links, e.g., e_{coding}, of node m_{coding}. Then, we divide all path sets $Paths(s, t_k)$, $k = 1, \ldots, d$, of $Solution(G_D)$ into two groups, i.e., *unaffectedPaths* and *affectedPaths*. Assume that there are h path sets in *affectedPaths*, where h is a positive integer smaller than the number of receivers d. So *unaffectedPaths* contains (d-h) path sets. $Paths(s, t_k)$ is included into *affectedPaths* if link $e_{coding} \in Paths(s, t_k)$; otherwise, $Paths(s, t_k)$ belongs to *unaffectedPaths*. Third, we reconstruct all path sets in *affectedPaths*, with *unaffectedPaths* unchanged. After that, all path sets in *unaffectedPaths* and *affectedPaths* are combined to form a new combined to form a new solution, aiming to reduce the coding operations involved.

The path set reconstruction is described below. First, with link e_{coding} unchanged, we delete the rest of the incoming links of node m_{coding} from G_D resulting in a new graph G'_D. Then, we send h ant groups to rebuild all path sets in *affectedPaths* over G'_D. Note that, it is possible that graph G'_D cannot meet the data rate requirement after the deletion of those incoming links. So, when rebuilding a path set, e.g., $Paths(s, t_k)$, we limit the number of times attempted. If the reconstruction cannot be completed after these attempts, NCRM-ACO gives up the reconstruction process; otherwise, replaces $Paths(s, t_k)$ with the newly constructed path set $Paths_{\text{new}}(s, t_k)$. After all path sets in *affectedPaths* are rebuilt or after a certain number of times, we combine *affectedPaths* with *unaffectedPaths* to form a new solution $Solution^{\text{new}}(G_D)$. If $Solution^{\text{new}}(G_D)$ requires less coding operations, $Solution(G_D)$ is replaced by $Solution^{\text{new}}(G_D)$. Otherwise, $Solution(G_D)$ remains unchanged.

For example, as shown in Fig. 4(d), there is only one coding node, i.e., node 4_o_1. Hence, the solution reconstruction method procedure starts with node 4_o_1. According to the procedure, we randomly choose an incoming link e of node 4_o_1, e.g., nodes 4_i_1 to 4_o_1. As link e is included in $Paths(1, 9)$ but not in $Paths(1, 8)$, we have $unaffectedPaths = \{Paths(1, 9)\}$ and $affectedPaths = \{Paths(1, 8)\}$. After that, apart from link e, the rest of the incoming links of node 4_o_1, i.e., link node 4_i_2 → node 4_o_1, is deleted from the graph, and the reconstruction of $Paths(1, 8)$ is triggered. Fig. 10(a) shows the new graph $G^{'}_D$ after the deletion of incoming links. Suppose the new ant group successfully constructs two link-disjoint paths, $p'_1(1, 8) = 1 \rightarrow 2 \rightarrow 8$ and $p'_2(1, 8) = 1 \rightarrow 3 \rightarrow 4\_i\_2 \rightarrow 4\_o\_2 \rightarrow 6 \rightarrow 7\_i\_2 \rightarrow 7\_o\_1 \rightarrow 8$. We thus have $Paths_{\text{new}}(1, 8) = \{p'_1(1, 8), p'_2(1, 8)\}$, as shown in Fig. 10(b). Then, a new solution is formed by combining $Paths_{\text{new}}(1, 8)$ and $Paths(1, 9)$, with no coding operation required [see Fig. 10(d)]. Due to $\varphi[G_{\text{NCM}}(Solution^{\text{new}}(G_D))] < \varphi[G_{\text{NCM}}(Solution(G_D))]$, we replace the old solution with $Solution^{\text{new}}(G_D)$.

G. Pheromone Global Updating Rule

In addition to the pheromone local updating rule, NCRM-ACO adopts a pheromone global updating rule to guide the search toward optimal solutions. Under this rule, the pheromone information on all links is updated by a historic best solution $Solution_{gb}(G_D)$, providing some instructive guidance to improve the quality of the solutions built. The pheromone value is updated by using

$$\tau(t_k, l, (i,j)) = (1-\rho)\tau(t_k, l, (i,j)) + \rho\Delta\tau_{gb} \quad (10)$$

$$\Delta\tau_{gb} = \begin{cases} \left(\varphi_{gb}\right)^{-1}, & \text{if } (i,j) \in Solution_{gb}(G_D) \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

where parameter $\rho \in (0, 1]$ is a constant value, called the evaporation rate, mimicking the evaporation of the pheromone on all links [22], i.e., the pheromone value on each link decreases by ρ whenever the global pheromone updating is executed. φ_{gb} is the number of coding nodes in $G_{\text{NCM}}(Solution_{gb}(G_D))$.

V. Performance Evaluation

In this section, we first introduce the test instances, the experimental environment and all metrics for performance evaluation. We then report an experiment which helps us to find a set of appropriate parameter values for NCRM-ACO. Later, we validate the effectiveness of all the proposed mechanisms of NCRM-ACO. Finally, the proposed algorithm is evaluated by comparing it against a number of state-of-the-art algorithms already developed for solving the NCRM problem.

A. Test Instances

We evaluate the performance of the proposed algorithm on 35 benchmark instances which can be classified into four categories, namely fixed, random, hybrid, and real-world networks. Table I shows all instances and their parameters. To encourage future scientific comparison on the NCRM problem, these instances are available at http://www.cs.nott.ac.uk/~rxq/benchmarks.htm. All experiments are run on a computer with Windows 8 OS, Intel Core i7-3740QM CPU 2.7 GHz and 8 GB RAM.

1) *Fixed Networks:* These four networks have been widely used in [8]–[10], [27]–[31], and [33]–[39]. They are also referred to as n-copy networks, each of which is built by cascading n copies of basic network [see Fig. 11(a)]. Fig. 11(c) illustrates the three-copy network, where node 1 is the source and nodes 16, 17, 24, and 25 are receivers. It can be easily inferred that the minimum number of the coding operations to any n-copy networks is 0.00.
2) *Random Networks:* Networks of this type are all generated by the directed acyclic graph generation method introduced in [46]. The 18 random networks have 20 to 500 nodes. It is noted that Rnd-11–Rnd-18 are relatively large networks.
3) *Hybrid Networks:* As all test cases have a global minimum of 0.00, we generated 8 hybrid networks, where the global minimum of each instance is at least 1 and is known beforehand. This is done by combining two basic networks together, i.e., Fig. 11(a) and (b), where Fig. 11(a) is coding-free while Fig. 11(b) has an explicit coding node, i.e., node 4. In this way, a hybrid network can be built by combining a number of Fig. 11(a) and (b) networks together. The global minimum of an instance is equal to the number of Fig. 11(b) networks. Therefore, in hybrid networks, the global minimum is already known. The hybrid networks are called X-hybrid(Y), where X represents the number of networks being combined and Y indicates the global minimum value. Similar to the 3-copy, 7-copy, 15-copy, and 31-copy networks, we create 3-hybrid, 7-hybrid, 15-hybrid, and 31-hybrid networks, respectively. The global minimum is from 1 to 5. Fig. 11(d) illustrates 3-hybrid(1) network which contains two Fig. 11(a) networks and one Fig. 11(b) network. The global minimum is 1. Therefore, hybrid networks could be used to simulate networks where coding is necessarily performed and reflect the optimization ability of the algorithm in solving this type of NCRM problem.
4) *Real-World Networks:* Five real-world topologies have been adopted for the performance evaluation, namely, Ebone-1, Ebone-2, Ebone-3, Exodus-1, and Exodus-2 [34], [35]. We also use them in our experiments.

B. Performance Measures

To thoroughly evaluate the performance of the proposed algorithm, the following performance measuring metrics are employed throughout the experiments.

1) Mean and standard deviation (SD) of the best solutions found from 50 runs. Mean and SD are important metrics to demonstrate the overall performance of a search algorithm.
2) Average computational time (ACT) consumed by an algorithm over 50 runs. This metric is a direct indication of the computational time of an algorithm.
3) Student's t-test [33], [47] to compare two algorithms (A and B) in terms of the objective function values of the 50 best solutions obtained. In this paper, two-tailed t-test with 98 degrees of freedom at a 0.05 level of significance is used. The t-test result can show statistically if the performance of A is better than, worse than, or equivalent to that of B.

C. Parameter Settings

The performance of the proposed ACO could be seriously deteriorated, e.g., leading to slow convergence and prematurity, if the values of parameters, namely the pheromone factor α, the heuristic factor β, the pheromone evaporation rate ρ, and the pseudorandom coefficient q_0, are inappropriately set. In order to determine an appropriate combination of the parameter values, for each parameter, we tested four possible values, i.e., $\alpha \in \{0.6, 0.7, 0.8, 0.9\}$, $\beta \in \{2, 3, 4, 5\}$, $\rho \in \{0.0, 0.1, 0.2, 0.3\}$, and $q_0 \in \{0.4, 0.5, 0.6, 0.7\}$. This may lead to $4^4 = 256$ combinations if we try all possible parameter values. However, it is not necessary to try all the combinations, since we only want to determine an appropriate combination,

TABLE I
EXPERIMENTAL INSTANCES AND THEIR PARAMETERS

Group	Networks	Original network G					Decomposed graph G_D			Optimum
		Nodes	Links	Receivers	Rate	Average degree	Nodes	Links	Input links for coding	
Fixed	3-copy	25	36	4	2	2.88	49	68	32	0
	7-copy	57	84	8	2	2.95	117	164	80	0
	15-copy	121	180	16	2	2.98	253	356	176	0
	31-copy	249	372	32	2	2.99	617	740	368	0
Random	Rnd-1	20	37	5	3	3.80	54	81	43	0
	Rnd-2	20	39	5	3	3.90	65	89	50	0
	Rnd-3	30	60	6	3	4.00	94	146	86	0
	Rnd-4	30	69	6	3	4.60	113	181	112	0
	Rnd-5	40	78	9	3	3.90	124	184	106	0
	Rnd-6	40	85	9	4	4.25	91	149	64	0
	Rnd-7	50	101	8	3	4.04	178	246	145	0
	Rnd-8	50	118	10	4	4.72	194	307	189	0
	Rnd-9	60	150	11	5	5.00	239	385	235	0
	Rnd-10	60	156	10	4	5.20	262	453	297	0
	Rnd-11	100	175	10	2	3.50	245	389	214	0
	Rnd-12	100	279	10	3	5.58	433	879	600	0
	Rnd-13	150	337	16	2	4.49	483	851	514	0
	Rnd-14	150	363	11	3	6.17	712	1519	1056	0
	Rnd-15	200	527	18	2	5.27	823	1586	1059	0
	Rnd-16	200	473	12	3	4.73	703	1272	799	0
	Rnd-17	500	1086	33	2	4.34	1682	2947	1861	0
	Rnd-18	500	491	24	3	5.46	2187	4413	3048	0
Hybrid	3-hybrid(1)	24	34	4	2	2.83	42	58	24	1
	3-hybrid(2)	23	32	4	2	2.78	35	48	16	2
	7-hybrid(2)	55	80	8	2	2.91	107	148	68	2
	7-hybrid(3)	54	78	8	2	2.89	102	140	62	3
	15-hybrid(3)	118	174	16	2	2.95	238	332	158	3
	15-hybrid(4)	117	172	16	2	2.94	233	324	152	4
	31-hybrid(4)	245	364	32	2	2.97	505	708	344	4
	31-hybrid(5)	244	362	32	2	2.97	500	700	338	5
Real world	Ebone-1	18	23	5	2	2.44	31	39	16	0
	Ebone-2	31	45	5	3	2.90	58	80	35	0
	Ebone-3	26	45	5	4	3.46	62	99	54	0
	Exodus-1	24	30	5	2	2.50	37	46	16	0
	Exodus-2	33	51	5	3	2.73	71	105	54	0

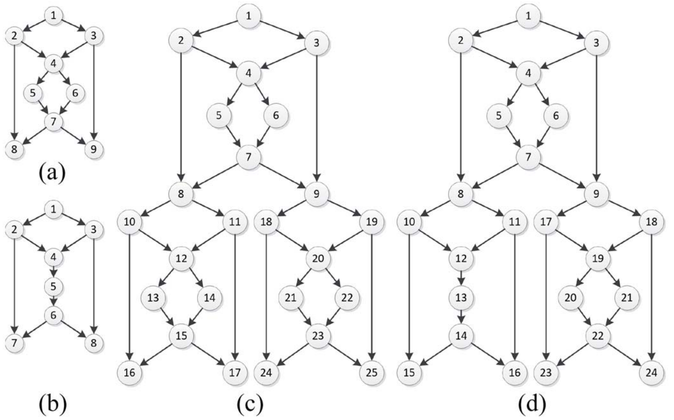


Fig. 11. Example of fixed and hybrid networks (a) Basic network 1. (b) Basic network 2. (c) 3-copy. (d) 3-hybrid(1).

rather than the best setting. We thus use the orthogonal experimental design (OED) to find a relatively good combination. OED is a multiparameter experimental design method based on orthogonal array, where a number of representative combinations of parameter values which are uniformly distributed within the test range are selected from the full parameter experiment [48]. This method is highly efficient when designing multiparameter experiments. It can greatly reduce the number of required experiments while obtaining promising results. Since its introduction in 1950s, OED has been widely applied in many areas, such as economic management, bioengineering, and environmental engineering, etc. [49]–[51]. The following briefly introduces the procedure of OED.

TABLE II
TABLE OF ORTHOGONAL ARRAY $L_{16}(4^4)$

ParaCom	α	β	ρ	q_0	ParaCom	α	β	ρ	q_0
1	1	1	1	1	9	3	1	3	4
2	1	2	2	2	10	3	2	4	3
3	1	3	3	3	11	3	3	1	2
4	1	4	4	4	12	3	4	2	1
5	2	1	2	3	13	4	1	4	2
6	2	2	1	4	14	4	2	3	1
7	2	3	4	1	15	4	3	2	4
8	2	4	3	2	16	4	4	1	3

Note: number x in the columns α, β, ρ, q_0 correspond to the x-th value in the parameter value set

Let $L_a(b^c)$ denote the orthogonal array, where a is the number of experiments, b is the levels of parameters, and c is the number of parameters. The orthogonal array has two properties, that is: 1) in each column, the number of occurrences of different numbers is equal and 2) in any two columns, the arrangement of numbers is complete and balanced. Any parameter at each level is thus compared to all different parameters with each other. Consequently, test results can be analyzed through range and variance analysis to determine a better value combination of parameters. More details can be found in [48]–[52]. In our experiment, an orthogonal array $L_{16}(4^4)$ is obtained from the referencing orthogonal table, where 16 representative combinations are listed in Table II.

We carry out 50 independent runs for each parameter combination and record the mean value of the best solutions. As Fix-4 network instance is one of the most difficult instances, we use it to run the parameter setting experiments.

TABLE III
RESULTS OF THE OED

ParaCom	α	β	ρ	q_0	Mean
1	0.6	2	0.0	0.4	6.82
2	0.6	3	0.1	0.5	5.78
3	0.6	4	0.2	0.6	0.60
4	0.6	5	0.3	0.7	1.28
5	0.7	2	0.1	0.6	0.56
6	0.7	3	0.0	0.7	4.40
7	0.7	4	0.3	0.4	0.62
8	0.7	5	0.2	0.5	1.16
9	0.8	2	0.2	0.7	1.04
10	0.8	3	0.3	0.6	0.00
11	0.8	4	0.0	0.5	0.64
12	0.8	5	0.1	0.4	1.18
13	0.9	2	0.3	0.5	2.86
14	0.9	3	0.2	0.4	1.62
15	0.9	4	0.1	0.7	0.44
16	0.9	5	0.0	0.6	1.84
m_1	3.62	2.82	3.43	2.56	/
m_2	1.69	2.95	1.99	2.61	/
m_3	0.72	0.58	1.11	0.75	/
m_4	1.69	1.37	1.19	1.79	/

Note: The symbol / means not applicable

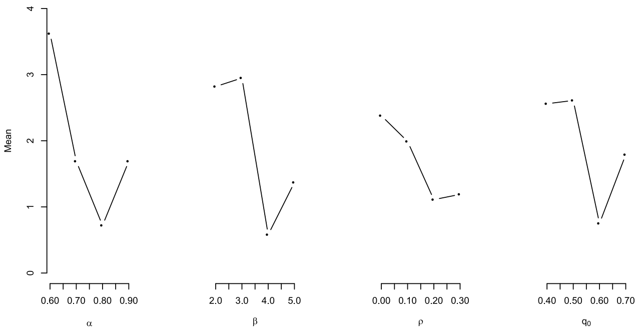


Fig. 12. Relationship between average time and parameters.

Table III shows the mean values of the 16 combinations in Table II. It is noted that rows m_1–m_4 represent the mean value of a certain parameter with a certain value. For instance, the mean value of parameter $\alpha = 0.6$ is calculated as (6.82+5.78+0.60+1.28)/4 = 3.62. So, value 3.62 is recorded in row m_1 and column α. Moreover, the mean value of each parameter is illustrated in Fig. 12. When $\alpha = 0.8$, $\beta = 4$, $\rho = 0.2$, and $q_0 = 0.6$, NCRM-ACO achieves the smallest mean value. Then, we compare the optimization performance of two combinations, i.e., $\{0.8, 4, 0.2, 0.6\}$ and $\{0.8, 3, 0.3, 0.6\}$ which gains the minimum mean value in Table II. In this experiment, performance indicators mean and ACT are used and the results are shown in Table IV. It is seen that each combination obtains a mean value of 0.00, indicating both of them can achieve the optimal solution in each single run. However, in terms of the ACT, ACO with $\{0.8, 4, 0.2, 0.6\}$ is faster. It is hence clear that the first combination in Table IV performs the best and is hereafter used as the parameter settings.

TABLE IV
RESULTS OF ADDITIONAL EXPERIMENTS

ParaCom	α	β	ρ	q_0	Mean	ACT (sec.)
1	0.8	4	0.2	0.6	0	14.84
2	0.8	3	0.3	0.6	0	18.72

D. *Effectiveness of the Proposed Mechanisms*

We evaluate the effectiveness of the proposed mechanisms by implementing two experiments on 14 selected instances, including the four fixed networks (3-copy, 7-copy, 15-copy, and 31-copy) and ten random networks (Rnd-1, ..., Rnd-10), because these instances have been widely used for performance evaluation.

The proposed ACO is featured with five specially devised mechanisms, including the multidimensional pheromone maintenance mechanism, the problem-specific heuristic information, the tabu-table-based path construction, the pheromone local updating rule, and the solution reconstruction method (see Section IV for details). Among them, the first two are essential components to drive ACO run properly. In other words, they are fundamental mechanisms that adapt ACO for the NCRM problem. One cannot test the effectiveness of the pheromone maintenance and the heuristic information in a separate way. Hence, we evaluate the two mechanisms as a whole (the first experiment) and test the others independently (the second experiment). The algorithms for comparison are listed below.

1) *Exp1:* Verification of the first two mechanisms.
   a) A1: The basic ACO [24].
   b) A2: A1 with the multidimensional pheromone maintenance mechanism and the problem-specific heuristic information (see Sections IV-B and IV-C).
2) *Exp2:* Independent verification of the remaining mechanisms.
   a) A3: A2 with the tabu-table-based path construction (see Section IV-D).
   b) A4: A2 with the pheromone local updating rule (see Section IV-E).
   c) A5: A2 with the tabu-table-based path construction and the pheromone local updating rule.
   d) A6: A2 with the solution reconstruction method (see Section IV-F).
   e) A7: A1 with all proposed mechanisms (also called NCRM-ACO).

As there is no clear heuristic information immediately available, we set heuristic information factor β of A1 to 0. For A2–A7, we set $\beta = 4$, with which the algorithm could achieve better optimization performance, as demonstrated in Section V-C.

Table V shows the experimental results of seven algorithms. First of all, it is easily observed that the basic ACO cannot build feasible solutions at all in any instance. The reason is as follows. As known, basic ACO utilizes a single pheromone table to guide the searching procedure of ants, which is in favor of addressing traditional path-finding problems such as traveling salesman problems [14], [21]. This is because, in the abovementioned path-finding problems, a single path is expected to be found between a source and a receiver. A single ant is able to finish the task. However, as for the NCRM problem, the task of an ant colony is to build a set of link-disjoint paths from the source to each receiver. Hence, a set of ants

TABLE V
RESULTS OF MEAN(SD) FOR THE SEVEN ALGORITHMS (BEST RESULTS ARE IN BOLD)

Network	Exp1		Exp2				
	A1	A2	A3	A4	A5	A6	A7
3-copy	/	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)
7-copy	/	1.62(1.50)	0.44(0.54)	0.48(0.62)	0.10(0.30)	0.86(0.72)	**0**(0)
15-copy	/	7.52(1.90)	4.72(1.08)	5.16(1.16)	4.54(1.19)	6.10(1.13)	**0**(0)
31-copy	/	21.84(5.67)	14.08(2.74)	15.18(2.52)	14.60(2.25)	18.90(2.62)	**0**(0)
Rnd-1	/	0.14(0.35)	**0**(0)	0.02(0.14)	**0**(0)	**0**(0)	**0**(0)
Rnd-2	/	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)
Rnd-3	/	0.28(0.43)	**0**(0)	0.06(0.24)	**0**(0)	**0**(0)	**0**(0)
Rnd-4	/	0.52(0.90)	**0**(0)	0.08(0.27)	**0**(0)	**0**(0)	**0**(0)
Rnd-5	/	1.50(1.27)	1.28(0.76)	1.38(1.31)	0.34(0.65)	0.26(1.22)	**0**(0)
Rnd-6	/	0.06(0.24)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)
Rnd-7	/	2.06(2.12)	0.76(0.48)	0.92(0.72)	0.54(0.50)	0.36(0.97)	**0**(0)
Rnd-8	/	2.72(2.53)	1.66(0.79)	1.92(1.34)	0.56(1.00)	0.24(0.81)	**0**(0)
Rnd-9	/	5.40(4.98)	1.36(1.38)	1.84(0.76)	0.46(1.19)	3.08(2.97)	**0**(0)
Rnd-10	/	2.92(2.04)	1.72(0.86)	2.04(1.22)	0.68(1.07)	1.54(1.82)	**0**(0)

Note: The symbol / stands for that the algorithm can't find any feasible solution

search in parallel to construct a number of link-disjoint paths. In this case, using a single pheromone table is far from enough to provide explicit guidance to each of the ants within the group, since the pheromone information on each link cannot simultaneously guide ants with different path-finding purposes. In addition, with no explicit heuristic information assisted, no local information can be utilized. Therefore, the basic ACO cannot even find feasible solutions in all instances. On the other hand, with the proposed pheromone maintenance mechanism and the heuristic information utilized, A2 is successfully applied to the NCRM problem.

Then, we compare A3, ..., A6 with A2 to verify if each of the three mechanisms has a positive impact on the performance of A2. Due to the nature of the NCRM problem, it is extremely difficult to find a satisfied set of link-disjoint paths. Hence, even with the pheromone maintenance mechanism and heuristic information integrated, it is still possible that an ant cannot reach its destination. In order to enhance the ability for an arbitrary ant to find a demanded path and diversify link-disjoint path sets, the tabu-table-based path construction is developed. It can be seen that A3 outperforms A2 in all instances in terms of mean value. Meanwhile, even if the tabu-table-based path construction is used, it is still possible to result into an infeasible path if inappropriate links are chosen. So, we need the pheromone local updating rule as a complement to the tabu-table-based path construction. If we compare the performance of A4 and A2, we see that the former is better. This is because, by using the punishing and rewarding schemes, the pheromone local updating rule is able to avoid ants following the same paths that the previous group has covered, which helps to improve the optimization performance. Meanwhile, by comparing the performance of A5, A4, and A3, we can verify the effectiveness of the tabu-table-based path construction and the pheromone local updating rule. Clearly, the two mechanisms perform better than any of them individually adopted. If looking at the results of A6 and A2, we also observe that the former performs better than the latter. The reason is explained below. As aforementioned, ACO may suffer from premature convergence. The solution reconstruction mechanism can improve the quality of solutions, so the local exploitation is enhanced and the local optimum can be avoided.

Finally, A7 is compared with A2, ..., A6. Obviously, A7 always obtain optimum in all of the instances. Regarding the mean and SD, it performs no worse, and usually better than the others. This demonstrates that equipped with all specially devised mechanisms, the proposed algorithm has a significantly improved optimization performance.

According to the abovementioned comparisons, we see that each of the proposed mechanisms contributes to the improvement of the NCRM-ACO. To further support our findings, we compare the seven algorithms using Student's *t*-test, where results are given in Table VI. The result of A↔B is shown as "+," "−," or "∼" when algorithm A is significantly better than, significantly worse than, or statistically equivalent to algorithm B, respectively. According to the results, A1 is beaten by A2 in all instances; A3–A6 outperform A2 in most of the instances; A5 performs better than A3 and A4; and A7 is the best algorithm among the seven. The results demonstrate not only the effectiveness of each mechanism but also the performance improvement via all the mechanisms.

E. Overall Performance Evaluation

We evaluate the overall performance of NCRM-ACO by comparing it with the 11 state-of-the-art EAs, including five BLS-based (BLSGA [10], QEA1 [28], QEA2 [27], PBIL [29], and cGA [31]); two BTS-based (BTSGA [10] and FA-ENCA [35]); three relative-encoding-based (RGA [36], SRHCGA [38], and CRO [39]); and one path-oriented (pEA [33]). The algorithms for performance comparison are listed as follows.

1) *BLSGA:* BLS encoding-based GA [10].
2) *QEA1:* Quantum-inspired evolutionary algorithm (QEA) [28].
3) *QEA2:* Another QEA proposed by Ji and Xing [27].
4) *PBIL:* Population-based incremental learning algorithm [29].
5) *cGA:* Compact genetic algorithm [31].
6) *BTSGA:* BTS encoding-based GA [10].
7) *FA-ENCA:* Fast and adaptive evolutionary algorithm [35].
8) *RGA:* GA proposed by Hu *et al.* [36].
9) *SRHCGA:* Spatial receding horizon control (SRHC) genetic algorithm [38].

TABLE VI
t-TEST RESULTS OF THE SEVEN ALGORITHMS

Network	A2↔A1	A3↔A2	A4↔A2	A5↔A2	A6↔A2	A7↔A2	A5↔A3	A5↔A4	A7↔A5	A7↔A6
3-copy	+	∼	∼	∼	∼	∼	∼	∼	∼	∼
7-copy	+	+	+	+	+	+	+	+	+	+
15-copy	+	+	+	+	+	+	+	+	+	+
31-copy	+	+	+	+	+	+	∼	+	+	+
Rnd-1	+	+	+	+	+	+	∼	+	∼	∼
Rnd-2	+	∼	∼	∼	∼	∼	∼	∼	∼	∼
Rnd-3	+	+	+	+	+	+	∼	+	∼	∼
Rnd-4	+	+	+	+	+	+	∼	+	∼	∼
Rnd-5	+	+	+	+	+	+	+	+	+	+
Rnd-6	+	+	+	+	+	+	∼	∼	∼	∼
Rnd-7	+	+	+	+	+	+	+	+	+	+
Rnd-8	+	+	+	+	+	+	+	+	+	+
Rnd-9	+	+	+	+	+	+	+	+	+	+

TABLE VII
RESULTS OF MEAN AND SD (BEST RESULTS ARE IN BOLD)

Network	BLSGA	QEA1	QEA2	PBIL	cGA	BTSGA	FA-ENCA	RGA	SRHCGA	CRO	pEA	NCRM-ACO
3-copy	0.52(0.84)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)
7-copy	2.36(2.22)	0.30(0.65)	0.74(1.18)	**0**(0)	**0**(0)	0.38(0.41)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)
15-copy	10.44(7.02)	2.82(3.31)	5.92(1.86)	1.76(2.57)	**0**(0)	0.08(0.22)	0.15(0.23)	**0**(0)	0.04(0.55)	0.10(0.41)	**0**(0)	**0**(0)
31-copy	31.66(6.48)	16.68(8.80)	20.02(0.22)	22.74(8.43)	**0**(0)	8.58(3.42)	20.35(3.90)	0.03(0.44)	0.01(0.14)	0.26(0.59)	**0**(0)	**0**(0)
Rnd-1	0.74(1.20)	0.12(0.31)	0.10(0.31)	**0**(0)	**0**(0)	0.28(0.44)	**0**(0)	0.01(0.14)	0.44(0.50)	0.20(0.40)	**0**(0)	**0**(0)
Rnd-2	0.26(0.64)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	0.02(0.22)	**0**(0)	**0**(0)	0.16(0.37)	0.06(0.47)	**0**(0)	**0**(0)
Rnd-3	0.24(0.68)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	0.08(0.27)	0.18(0.48)	**0**(0)	**0**(0)
Rnd-4	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)
Rnd-5	1.42(0.88)	0.46(0.51)	0.38(0.41)	**0**(0)	0.12(0.36)	0.38(0.57)	**0**(0)	0.50(0.50)	0.58(0.58)	0.96(0.52)	**0**(0)	**0**(0)
Rnd-6	0.22(0.41)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	0.06(0.47)	0.46(0.81)	0.48(0.59)	**0**(0)	**0**(0)
Rnd-7	1.38(0.97)	0.72(0.57)	0.62(0.48)	0.28(0.41)	0.56(0.58)	0.58(0.51)	**0**(0)	0.80(1.41)	0.12(0.69)	1.70(1.42)	**0**(0)	**0**(0)
Rnd-8	2.54(2.08)	0.78(0.85)	0.72(0.71)	0.32(0.31)	0.38(0.51)	0.92(0.56)	**0**(0)	1.40(1.56)	0.50(1.03)	2.10(1.53)	**0**(0)	**0**(0)
Rnd-9	2.76(1.25)	1.58(1.05)	1.58(0.99)	**0**(0)	0.12(0.41)	0.88(0.63)	**0**(0)	1.26(0.76)	0.72(0.72)	2.54(1.53)	**0**(0)	**0**(0)
Rnd-10	3.18(2.67)	0.48(0.68)	0.28(0.47)	0.04(0.22)	0.08(0.22)	0.96(0.59)	**0**(0)	1.78(1.96)	0.52(0.76)	2.82(2.33)	**0**(0)	**0**(0)
Rnd-11	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	0.12(0.33)	**0**(0)	**0**(0)
Rnd-12	0.28(0.57)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	0.24(0.44)	**0**(0)	0.52(0.50)	0.14(0.35)	0.64(0.48)	**0**(0)	**0**(0)
Rnd-13	25.32(25.30)	0.02(0.22)	**0**(0)	**0**(0)	**0**(0)	0.18(0.37)	**0**(0)	0.48(0.59)	0.16(0.73)	0.68(1.07)	**0**(0)	**0**(0)
Rnd-14	25.20(25.45)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	0.14(0.37)	**0**(0)	0.22(0.41)	0.30(1.01)	0.78(0.97)	**0**(0)	**0**(0)
Rnd-15	0.16(0.37)	0.02(0.22)	0.20(0.52)	**0**(0)	**0**(0)	0.08(0.22)	**0**(0)	0.36(0.72)	0.90(1.92)	0.60(0.80)	**0**(0)	**0**(0)
Rnd-16	2.34(1.35)	1.30(0.92)	1.48(0.89)	0.40(1.57)	**0**(0)	1.24(0.91)	**0**(0)	1.90(1.37)	1.48(1.15)	2.54(1.98)	**0**(0)	**0**(0)
Rnd-17	1.72(1.42)	0.84(0.90)	1.01(0.47)	1.08(0.29)	**0**(0)	1.40(0.75)	0.20(0.40)	5.68(6.01)	5.38(3.85)	7.14(4.15)	**0**(0)	**0**(0)
Rnd-18	8.26(2.59)	2.04(0.90)	1.18(0.37)	1.20(0.40)	**0**(0)	9.50(1.24)	1.30(1.08)	10.20(2.34)	7.38(1.49)	12.45(3.35)	**0**(0)	**0**(0)
3-hybrid(1)	1.16(0.37)	**1**(0)	**1**(0)	1.16(0.49)	**1**(0)	1.04(0.22)	**1**(0)	**1**(0)	**1**(0)	**1**(0)	**1**(0)	**1**(0)
3-hybrid(2)	2.22(0.44)	**2**(0)	**2**(0)	**2**(0)	**2**(0)	**2**(0)	**2**(0)	**2**(0)	**2**(0)	**2**(0)	**2**(0)	**2**(0)
7-hybrid(2)	3.66(1.39)	2.10(0.31)	3(2.13)	2.54(1.23)	**2**(0)	2.40(0.60)	2.10(0.30)	**2**(0)	3.20(1.79)	2.10(0.30)	**2**(0)	**2**(0)
7-hybrid(3)	4.98(2.48)	3.10(0.31)	3.60(1.85)	3.88(1.84)	**3**(0)	3.22(0.44)	3.34(2.24)	**3**(0)	3.50(1.09)	3.34(2.24)	**3**(0)	**3**(0)
15-hybrid(3)	10.70(5.44)	6.20(4.49)	8.44(3.80)	5.76(3.85)	3.70(0.47)	4.72(0.92)	4.64(1.38)	7.32(3.47)	5.46(4.82)	8.30(3.67)	**3**(0)	**3**(0)
15-hybrid(4)	11.12(4.88)	10.90(8.33)	8.80(4.16)	7.34(4.83)	**4**(0)	5.30(1.08)	5.64(2.58)	9.48(3.58)	5.72(4.50)	9.64(3.46)	**4**(0)	**4**(0)
31-hybrid(4)	37(9.27)	31.70(13.26)	28.06(8.94)	37.10(10.90)	**4**(0)	10.90(2.64)	10.20(1.56)	30.80(7.95)	26.80(11.93)	35.80(9.18)	**4**(0)	**4**(0)
31-hybrid(5)	32.80(8.12)	30.94(14.90)	28.20(3.88)	29.60(11.82)	**5**(0)	10.90(1.29)	9.80(3.23)	33.40(7.18)	25.50(12.38)	36.20(8.92)	**5**(0)	**5**(0)
Ebone-1	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)
Ebone-2	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)
Ebone-3	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)
Exodus-1	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)
Exodus-2	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)	**0**(0)

10) *CRO:* Chemical reaction optimization algorithm [39].
11) *pEA:* The path-oriented encoding EA [33].
12) *NCRM-ACO:* The proposed algorithm.

The population size is set to 20 and the maximum number of generations is 200 for each EA. For BLSGA, we set the crossover probability $p_c = 0.8$ and the mutation probability $p_m = 0.006$. For BTSGA, we have $p_c = 0.8$ and $p_m = 0.012$. For the rest of the algorithms, we adopt their best parameter settings in [27], [28], [30], [31], [33], [35], [36], [38], and [39]. For the fixed, random and real-world networks, the stopping criterion is either an optimal solution is obtained or the maximum number of generations is reached. For the hybrid networks, an algorithm stops when either the best-so-far solution has not been changed over 20 generations or the maximum number of generations is reached. The results of mean and SD are collected in Table VII, where the value should read mean (SD). Tables VIII and IX illustrate the *t*-test results and the ACTs of the 12 algorithms.

First of all, we compare the performance of algorithms based on the same encoding approach. Among the five BLS-based EAs, cGA gains the best overall performance. It is able to obtain the minimum mean value in almost all instances. As the optimal solution to each instance is already known, cGA obtains the optimal solution in each run in 29 instances. This is because cGA adopts a local search mechanism that exploits the local information of the underlying problem to locate promising areas and solutions. Regarding the BTS-based algorithms, BTSGA is beaten by FA-ENCA in 21 instances while the former wins four instances. Compared with BTSGA, FA-ENCA has a more stable performance, especially in small-scale networks. This is due to the self-adaptive fitness assignment rule and the entropy-based relaxation technique introduced in FA-ENCA.

TABLE VIII
t-TEST RESULTS FOR THE 12 ALGORITHMS

Network	3-copy	7-copy	15-copy	31copy	Rnd-1	Rnd-2	Rnd-3
NCRM-ACO↔BLSGA	+	+	+	+	+	+	+
NCRM-ACO↔QEA1	∼	+	+	+	+	∼	∼
NCRM-ACO↔QEA2	∼	+	+	+	+	∼	∼
NCRM-ACO↔PBIL	∼	∼	+	+	∼	∼	∼
NCRM-ACO↔cGA	∼	∼	∼	∼	∼	∼	∼
NCRM-ACO↔BTSGA	∼	+	+	+	+	+	∼
NCRM-ACO↔FA-ENCA	∼	∼	+	+	∼	∼	∼
NCRM-ACO↔RGA	∼	∼	∼	+	+	∼	∼
NCRM-ACO↔SRHCGA	∼	∼	+	+	+	+	+
NCRM-ACO↔CRO	∼	∼	+	+	+	+	+
NCRM-ACO↔pEA	∼	∼	∼	∼	∼	∼	∼
	Rnd-4	Rnd-5	Rnd-6	Rnd-7	Rnd-8	Rnd-9	Rnd-10
NCRM-ACO↔BLSGA	∼	+	+	+	+	+	+
NCRM-ACO↔QEA1	∼	+	∼	+	+	+	+
NCRM-ACO↔QEA2	∼	+	∼	+	+	+	+
NCRM-ACO↔PBIL	∼	∼	∼	+	+	∼	+
NCRM-ACO↔cGA	∼	+	∼	+	+	+	+
NCRM-ACO↔BTSGA	∼	+	∼	+	+	+	+
NCRM-ACO↔FA-ENCA	∼	∼	∼	∼	∼	∼	∼
NCRM-ACO↔RGA	∼	+	+	+	+	+	+
NCRM-ACO↔SRHCGA	∼	+	+	+	+	+	+
NCRM-ACO↔CRO	∼	+	+	+	+	+	+
NCRM-ACO↔pEA	∼	∼	∼	∼	∼	∼	∼
	Rnd-11	Rnd-12	Rnd-13	Rnd-14	Rnd-15	Rnd-16	Rnd-17
NCRM-ACO↔BLSGA	∼	+	+	+	+	+	+
NCRM-ACO↔QEA1	∼	∼	+	∼	+	+	+
NCRM-ACO↔QEA2	∼	∼	∼	∼	+	+	+
NCRM-ACO↔PBIL	∼	∼	∼	∼	∼	+	+
NCRM-ACO↔cGA	∼	∼	∼	∼	∼	∼	∼
NCRM-ACO↔BTSGA	∼	+	+	+	+	+	+
NCRM-ACO↔FA-ENCA	∼	∼	∼	∼	∼	∼	+
NCRM-ACO↔RGA	∼	+	+	+	+	+	+
NCRM-ACO↔SRHCGA	∼	+	+	+	+	+	+
NCRM-ACO↔CRO	+	+	+	+	+	+	+
NCRM-ACO↔pEA	∼	∼	∼	∼	∼	∼	∼
	Rnd-18	3-hybrid(1)	3-hybrid(2)	7-hybrid(2)	7-hybrid(3)	15-hybrid(3)	15-hybrid(4)
NCRM-ACO↔BLSGA	+	+	+	+	+	+	+
NCRM-ACO↔QEA1	+	∼	∼	+	+	+	+
NCRM-ACO↔QEA2	+	∼	∼	+	+	+	+
NCRM-ACO↔PBIL	+	+	∼	+	+	+	+
NCRM-ACO↔cGA	∼	∼	∼	∼	∼	∼	∼
NCRM-ACO↔BTSGA	+	+	∼	+	+	+	+
NCRM-ACO↔FA-ENCA	+	∼	∼	+	+	+	+
NCRM-ACO↔RGA	+	∼	∼	∼	∼	+	+
NCRM-ACO↔SRHCGA	+	∼	∼	+	+	+	+
NCRM-ACO↔CRO	+	∼	∼	+	+	+	+
NCRM-ACO↔pEA	∼	∼	∼	∼	∼	∼	∼
	31-hybrid(4)	31-hybrid(5)	Ebone-1	Ebone-2	Ebone-3	Exodus-1	Exodus-2
NCRM-ACO↔BLSGA	+	+	∼	∼	∼	∼	∼
NCRM-ACO↔QEA1	+	+	∼	∼	∼	∼	∼
NCRM-ACO↔QEA2	+	+	∼	∼	∼	∼	∼
NCRM-ACO↔PBIL	+	+	∼	∼	∼	∼	∼
NCRM-ACO↔cGA	∼	∼	∼	∼	∼	∼	∼
NCRM-ACO↔BTSGA	+	+	∼	∼	∼	∼	∼
NCRM-ACO↔FA-ENCA	+	+	∼	∼	∼	∼	∼
NCRM-ACO↔RGA	+	+	∼	∼	∼	∼	∼
NCRM-ACO↔SRHCGA	+	+	∼	∼	∼	∼	∼
NCRM-ACO↔CRO	+	+	∼	∼	∼	∼	∼
NCRM-ACO↔pEA	∼	∼	∼	∼	∼	∼	∼

Note: The result of comparison between algorithm A and B is shown as '+', '-', or '∼' when the former is significantly better than, significantly worse than, or statistically equivalent to the latter, respectively.

Looking at those with relative-encoding, RGA wins 11 instances and SRHCGA wins 15 out of all instances. The performance of RGA is excellent in small-scale instances but deteriorates with the increasing network scale. This is because the individuals become increasingly more complicated with the growth of network size and it is more difficult to satisfy the expected data rate in larger instances. SRHCGA is a constructive algorithm, where GA is integrated into the solution construction procedure. Due to the inherent short-sighted effect, SRHCGA cannot perform well in large-scale networks.

If comparing all algorithms, NCRM-ACO and pEA gain the best performance. They both achieve the best mean and SD values in all instances, meaning that optimal solutions are always found. As reported in [33], pEA is one of the best optimization algorithms for the NCRM problem. NCRM-ACO performs no worse than pEA, which indicates our proposed algorithm achieves a decent performance. This is mainly because a number of the problem-specific mechanisms have been integrated into the framework of ACO to enhance its overall performance. These mechanisms include the multidimensional pheromone maintenance mechanism

TABLE IX
RESULTS OF ACT (S) (BEST RESULTS ARE IN BOLD)

Network	BLSGA	QEA1	QEA2	PBIL	cGA	BTSGA	FA-ENCA	RGA	SRHCGA	CRO	pEA	NCRM-ACO
3-copy	0.92	0.17	0.23	0.10	0.02	0.82	0.04	0.06	2.90	0.35	0.06	**0.01**
7-copy	7.26	6.56	10.64	1.44	0.11	8.26	2.01	1.16	6.06	1.31	0.21	**0.04**
15-copy	32.31	59.73	66.52	51.23	1.52	150.80	78.70	24.94	63.94	6.72	1.17	**0.62**
31-copy	143.02	514.00	508.62	431.15	21.24	333.70	332.32	288.35	316.82	28.32	15.72	**14.84**
Rnd-1	2.22	1.03	0.97	0.21	0.23	2.12	0.06	0.57	3.10	0.66	0.13	**0.03**
Rnd-2	1.14	0.39	0.37	0.12	0.02	1.01	0.05	0.08	4.19	0.26	0.10	**0.01**
Rnd-3	3.01	0.44	0.49	0.13	0.05	4.05	0.11	0.63	7.45	1.30	0.21	**0.01**
Rnd-4	2.07	0.47	0.56	0.16	0.07	1.83	0.13	2.45	4.40	1.67	0.15	**0.01**
Rnd-5	10.56	10.64	7.67	5.18	2.38	17.71	2.28	5.19	12.32	8.87	0.58	**0.54**
Rnd-6	1.84	0.44	0.63	0.11	0.03	2.03	0.12	6.31	16.54	4.74	0.19	**0.02**
Rnd-7	14.60	17.12	21.32	10.54	3.80	31.09	2.05	5.12	26.20	2.65	1.70	**0.33**
Rnd-8	25.10	20.82	25.26	15.94	7.10	52.31	5.09	18.08	79.75	4.01	0.65	**0.03**
Rnd-9	37.31	47.36	48.91	37.33	8.86	122.20	22.55	30.88	96.37	7.02	2.46	**1.18**
Rnd-10	39.69	31.82	52.43	22.39	9.19	264.82	11.47	41.19	95.62	12.08	0.81	**0.58**
Rnd-11	9.91	2.69	2.67	0.05	0.03	6.29	0.58	23.86	105.62	6.18	0.40	**0.03**
Rnd-12	70.44	11.75	14.95	2.40	0.39	65.94	1.69	21.26	113.34	37.81	1.40	**0.23**
Rnd-13	133.72	51.78	37.85	3.57	0.55	138.62	11.06	67.10	147.81	31.93	2.53	**0.28**
Rnd-14	284.57	38.31	34.68	3.66	2.06	239.84	10.09	96.12	172.06	167.51	3.53	**1.16**
Rnd-15	610.61	193.65	250.50	16.49	2.56	468.72	31.04	229.93	384.09	181.80	8.44	**1.18**
Rnd-16	305.03	299.60	362.34	165.63	43.70	286.69	113.24	439.76	423.75	74.30	18.12	**4.48**
Rnd-17	5432.30	6224.80	6250.50	2989.70	397.20	6070.28	2859.75	3948.10	1523.86	1064.61	61.61	**11.31**
Rnd-18	10635.40	9529.90	9215.77	6319.60	1914.75	9553.05	9984.34	10238.23	1689.62	4032.95	83.90	**51.35**
3-hybrid(1)	1.88	4.20	2.65	8.57	0.06	1.38	0.22	0.15	4.79	0.77	5.23	**0.05**
3-hybrid(2)	1.84	3.65	2.42	7.98	0.08	1.39	0.21	0.11	4.86	0.83	5.52	**0.05**
7-hybrid(2)	3.74	12.76	6.76	20.99	**0.27**	5.39	2.58	3.10	10.11	1.53	19.07	0.30
7-hybrid(3)	4.40	11.22	6.23	20.54	0.34	4.94	2.34	3.51	9.94	1.45	17.38	**0.33**
15-hybrid(3)	22.02	56.46	39.80	79.12	**3.46**	23.62	18.14	8.52	60.89	3.18	107.60	3.76
15-hybrid(4)	23.55	42.11	40.82	81.20	**2.71**	24.79	18.22	9.82	62.47	3.25	131.67	4.00
31-hybrid(4)	114.58	224.09	241.59	364.92	**22.07**	151.65	91.79	23.94	101.67	14.68	2082.70	51.38
31-hybrid(5)	114.78	206.78	230.61	311.80	**20.55**	156.44	74.83	20.62	128.92	13.92	2707.60	55.33
Ebone-1	0.011	0.009	0.013	0.010	**0.005**	0.013	0.022	0.072	1.072	0.049	0.020	0.007
Ebone-2	0.170	0.041	0.038	0.024	**0.014**	0.172	0.027	0.157	1.206	0.108	0.038	0.016
Ebone-3	0.065	0.014	0.013	0.020	**0.004**	0.071	0.021	0.056	0.983	0.088	0.029	0.011
Exodus-1	0.036	0.016	0.011	0.013	**0.009**	0.031	0.057	0.022	1.581	0.023	0.054	0.045
Exodus-2	0.738	0.114	0.138	0.015	**0.006**	0.209	0.028	0.437	1.347	0.361	0.041	0.010

which eliminates the pheromone overlapping phenomenon, the heuristic maintenance mechanism which exploits the local information to provide extra guidance to reduce the number of coding operations in the solution construction process, the tabu-table-based path construction and the pheromone local updating rule for easily and properly finding feasible paths connecting the source and each receiver, and the solution reconstruction method which improves the exploitation ability of ACO. With all the abovementioned mechanisms, NCRM-ACO performs well when tackling the NCRM problem. To further support our analysis, we compare the 12 algorithms using Student's *t*-test. Obviously, NCRM-ACO and pEA are the two best algorithms among all algorithms for comparison.

Then, we compare the ACTs obtained by different algorithms. NCRM-ACO is one of the fastest in almost all instances. The following explains the reasons. Different from the existing algorithms being compared, NCRM-ACO is based on the principle learning while optimizing. With all the problem-specific mechanisms integrated, NCRM-ACO makes use of the local and global information collected during the search so as to guide the fast construction of optimal solutions. Hence, less computational time is consumed. For relatively small instances, such as Rnd-3 and Rnd-8, NCRM-ACO is 20 times faster compared to pEA, the second fastest algorithm. For large instances, e.g., Fix-4, Rnd-17, and Rnd-18, although constructing feasible paths may waste some time, NCRM-ACO is still able to obtain an optimal solution within a very limited time, i.e., the fastest one among those algorithms being compared. As compared above, NCRM-ACO and pEA both gain the best performance with respect to the best solutions obtained. However, if looking at the ACT indicator, one can easily see that NCRM-ACO is much faster than pEA in almost all instances. When considering the practical deployment of the NCM, the computational time is of vital importance since the algorithm needs to respond to applications as quickly as possible. So, if we take into account mean, SD, the *t*-test results, and ACT, NCRM-ACO has the best overall performance and is definitely better than pEA, our previous work.

VI. CONCLUSION

This paper proposed a modified NCRM-ACO algorithm based on ACO to tackle the NCRM problem. Different from the existing algorithms, NCRM-ACO constructs feasible solutions with the help of the local and global information that emerged during the search. The proposed algorithm has several attractive features which contribute to its descent performance. Instead of using a single pheromone table, multiple pheromone tables are maintained in the pheromone maintenance mechanism so that each ant is appropriately guided to complete its path-finding task. The problem-specific heuristic information exposes the status of each incoming link to ant groups. Thus, each ant is provided with useful local information for selecting appropriate links along the path under construction. A tabu-table-based path construction mechanism and a pheromone local updating rule are devised to achieve a higher successful ratio for constructing feasible path sets. Moreover, a solution reconstruction method is able to enhance the local exploration ability of NCRM-ACO, with the purpose of improving the solution quality. With these problem-specific mechanisms integrated, NCRM-ACO is reported to outperform seven existing state-of-the-art algorithms in terms of the best solutions obtained and the ACT on a set of widely tested benchmark problems.

Acknowledgment

The authors would like to thank Dr. C. W. Ahn from Sungkyunkwan University, Korea, for kindly providing source code and test instances of his previous work on multicasting with network coding.

References

[1] C. Fragouli and E. Soljanin, *Network Coding Fundamentals*. Boston, MA, USA: Now, 2007.

[2] L. R. Ford and D. R. Fulkerson, “Maximal flow through a network,” *Can. J. Math.*, vol. 8, no. 3, pp. 399–404, 1956.

[3] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, “Network information flow,” *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[4] S.-Y. R. Li, R. W. Yeung, and N. Cai, “Linear network coding,” *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[5] R. Koetter and M. Médard, “An algebraic approach to network coding,” *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 782–795, Oct. 2003.

[6] S. Jaggi *et al.*, “Polynomial time algorithms for multicast network code construction,” *IEEE Trans. Inf. Theory*, vol. 51, no. 6, pp. 1973–1982, Oct. 2005.

[7] P. A. Chou and Y. Wu, “Network coding for the Internet and wireless networks,” *IEEE Signal Process. Mag.*, vol. 24, no. 5, pp. 77–85, Sep. 2007.

[8] M. Kim, C. W. Ahn, M. Médard, and M. Effros, “On minimizing network coding resources: An evolutionary approach,” in *Proc. 2nd Workshop Netw. Coding, Theory, Appl. (NetCod)*. Boston, MA, USA, 2006.

[9] M. Kim *et al.*, “Evolutionary approaches to minimizing network coding resources,” in *Proc. IEEE Int. Conf. Comput. Commun.*, Anchorage, AK, USA, 2007, pp. 1991–1999.

[10] M. Kim, V. Aggarwal, U.-M. O’Reilly, M. Médard, and W. Kim, “Genetic representations for evolutionary minimization of network coding resources,” in *Applications of Evolutionary Computing*. Berlin, Germany: Springer, 2007, pp. 21–31.

[11] M. Langberg, A. Sprintson, and J. Bruck, “The encoding complexity of network coding,” *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2386–2397, Jun. 2006.

[12] R. Battiti, M. Brunato, and F. Mascia, “Reactive search and intelligent optimization,” Dept. Inf. Commun. Technol., Univ. Trento, Trento, Italy, Tech. Rep. DIT-07-049, 2007.

[13] R. Battiti and G. Tecchiolli, “The reactive tabu search,” *ORSA J. Comput.*, vol. 6, no. 2, pp. 126–140, 1994.

[14] M. Dorigo and L. M. Gambardella, “Ant colony system: A cooperative learning approach to the traveling salesman problem,” *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, Apr. 1997.

[15] T. Stützle and H. H. Hoos, “Max-min ant system,” *Future Gener. Comput. Syst.*, vol. 16, no. 8, pp. 889–914, 2000.

[16] O. Cordon, I. F. de Viana, F. Herrera, and L. Moreno, “A new ACO model integrating evolutionary computation concepts: The best-worst ant system,” in *Proc. 2nd Int. Workshop Ant Algorithms*, Brussels, Belgium, 2000, pp. 22–29.

[17] J. E. Bell and P. R. McMullen, “Ant colony optimization techniques for the vehicle routing problem,” *Adv. Eng. Inform.*, vol. 18, no. 1, pp. 41–48, 2004.

[18] T. Stützle and M. Dorigo, “ACO algorithms for the quadratic assignment problem,” in *New Ideas Optimization*, Maidenhead, U.K.: McGraw-Hill, 1999, pp. 33–50.

[19] D. Merkle, M. Middendorf, and H. Schmeck, “Ant colony optimization for resource-constrained project scheduling,” *IEEE Trans. Evol. Comput.*, vol. 6, no. 4, pp. 333–346, Aug. 2002.

[20] T. Liao, K. Socha, M. A. M. de Oca, T. Stützle, and M. Dorigo, “Ant colony optimization for mixed-variable optimization problems,” *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 503–518, Aug. 2014.

[21] M. Dorigo and L. M. Gambardella, “Ant colonies for the travelling salesman problem,” *BioSystems*, vol. 43, no. 2, pp. 73–81, 1997.

[22] M. Dorigo and C. Blum, “Ant colony optimization theory: A survey,” *Theor. Comput. Sci.*, vol. 344, no. 2, pp. 243–278, 2005.

[23] M. Dorigo, M. Birattari, and T. Stützle, “Ant colony optimization,” *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, Nov. 2006.

[24] M. Dorigo and M. Birattari, “Ant colony optimization,” in *Encyclopedia of Machine Learning*. New York, NY, USA: Springer, 2010, pp. 36–39.

[25] C. Fragouli and E. Soljanin, “Information flow decomposition for network coding,” *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 829–848, Mar. 2006.

[26] H. Xing, R. Qu, L. Bai, and Y. Ji, “On minimizing coding operations in network coding based multicast: An evolutionary algorithm,” *Appl. Intell.*, vol. 41, no. 3, pp. 820–836, 2014.

[27] Y. Ji and H. Xing, “A memory-storable quantum-inspired evolutionary algorithm for network coding resource minimization,” in *Proc. Evolutionary Algorithms (InTech)*, Shanghai, China, 2011, pp. 363–380.

[28] H. Xing, Y. Ji, L. Bai, and Y. Sun, “An improved quantum-inspired evolutionary algorithm for coding resource optimization based network coding multicast scheme,” *AEU Int. J. Electr. Commun.*, vol. 64, no. 12, pp. 1105–1113, 2010.

[29] H. Xing and R. Qu, “A population based incremental learning for network coding resources minimization,” *IEEE Commun. Lett.*, vol. 15, no. 7, pp. 698–700, Jul. 2011.

[30] H. Xing and R. Qu, “A population based incremental learning for delay constrained network coding resource minimization,” in *Applications of Evolutionary Computation*. Berlin, Germany: Springer, 2011, pp. 51–60.

[31] H. Xing and R. Qu, “A compact genetic algorithm for the network coding based resource minimization problem,” *Appl. Intell.*, vol. 36, no. 4, pp. 809–823, 2012.

[32] H. Xing and R. Qu, “A nondominated sorting genetic algorithm for bi-objective network coding based multicast routing problems,” *Inf. Sci.*, vol. 233, pp. 36–53, Jun. 2013.

[33] H. Xing, R. Qu, G. Kendall, and R. Bai, “A path-oriented encoding evolutionary algorithm for network coding resource minimization,” *J. Oper. Res. Soc.*, vol. 65, no. 8, pp. 1261–1277, 2013.

[34] C. W. Ahn and M. Kim, “Self-adaptive evolutionary network coding algorithm: A constraint handling approach,” in *Proc. IEEE 3rd Int. Workshop Adv. Comput. Intell. (IWACI)*, Suzhou, China, 2010, pp. 238–243.

[35] C. W. Ahn, “Fast and adaptive evolutionary algorithm for minimum-cost multicast with network coding,” *Electr. Lett.*, vol. 47, no. 12, pp. 700–701, Jun. 2011.

[36] X. B. Hu, M. S. Leeson, and E. L. Hines, “An effective genetic algorithm for network coding,” *Comput. Oper. Res.*, vol. 39, no. 5, pp. 952–963, 2012.

[37] X. B. Hu, M. S. Leeson, and E. L. Hines, “A ripple-spreading genetic algorithm for the network coding problem,” in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Barcelona, Spain, 2010, pp. 1–8.

[38] X. B. Hu and M. S. Leeson, “Evolutionary computation with spatial receding horizon control to minimize network coding resources,” *Sci. World J.*, vol. 2014, Apr. 2014, Art. ID 268152.

[39] B. Pan, A. Y. S. Lam, and V. O. K. Li, “Network coding optimization based on chemical reaction optimization,” in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Houston, TX, USA, 2011, pp. 1–5.

[40] M. Kim, M. Médard, V. Aggarwal, and U.-M. O’Reilly, “On the coding-link cost tradeoff in multicast network coding,” in *Proc. IEEE Mil. Comun. Conf. (MILCOM)*, Orlando, FL, USA, 2007, pp. 1–7.

[41] M. Kim, M. Médard, and U.-M. O’Reilly, “Network coding and its implications on optical networking,” in *Proc. Conf. Optical Fiber Comun. (OFC)*, San Diego, CA, USA, 2009, pp. 1–3.

[42] C. W. Ahn and J.-C. Yoo, “Multi-objective evolutionary approach to coding-link cost trade-offs in network coding,” *Electr. Lett.*, vol. 48, no. 25, pp. 1595–1596, Dec. 2012.

[43] L. P. Karunarathne, M. S. Leeson, and E. L. Hines, “Evolutionary minimization of network coding resources,” *Appl. Artif. Intell.*, vol. 28, no. 9, pp. 837–858, 2014.

[44] Z. Ren, Z. Feng, and A. Zhang, “Fusing ant colony optimization with Lagrangian relaxation for the multiple-choice multidimensional knapsack problem,” *Inf. Sci.*, vol. 182, no. 1, pp. 15–29, 2012.

[45] B. Bullnheimer, R. F. Hartl, and C. Strauss, "A new rank based version of the ant system—A computational study," *Cent. Eur. J. Oper. Res. Econ.*, vol. 7, pp. 25–38, Apr. 1997.

[46] G. Melançon and F. Philippe, "Generating connected acyclic digraphs uniformly at random," *Inf. Process. Lett.*, vol. 90, no. 4, pp. 209–213, 2004.

[47] R. E. Walpole, R. H. Myers, S. L. Myers, and K. Ye, *Probability and Statistics for Engineers and Scientists*, vol. 5. New York, NY, USA: Macmillan, 1993.

[48] G. Taguchi, *System of Experimental Design: Engineering Methods to Optimize Quality and Minimize Costs.-3. Pr*. White Plains, NY, USA: UNIPUB/Kraus, 1988.

[49] M. Abud-Archila *et al.*, "Optimization of osmotic dehydration of yam bean (pachyrhizus erosus) using an orthogonal experimental design," *J. Food Eng.*, vol. 84, no. 3, pp. 413–419, 2008.

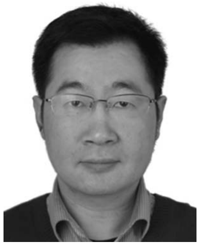
[50] W. Cui, X. Li, S. Zhou, and J. Weng, "Investigation on process parameters of electrospinning system through orthogonal experimental design," *J. Appl. Polymer Sci.*, vol. 103, no. 5, pp. 3105–3112, 2007.

[51] J. Zhu, D. A. Chew, S. Lv, and W. Wu, "Optimization method for building envelope design to minimize carbon emissions of building operational energy consumption using orthogonal experimental design (OED)," *Habitat Int.*, vol. 37, pp. 148–154, Jan. 2013.

[52] K. Fang and C. Ma, *Orthogonal and Uniform Experimental Design*. Beijing, China: Sci. Technol. Press, 2001.

Zhaoyuan Wang received the B.S. degree in computer software from the School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China, in 2013, where he is currently working toward the M.S. degree in software engineering.

His research interests include ant colony optimization, multiobjective optimization, network coding, and cloud computing.



Huanlai Xing received the B.Sc. (Eng.) degree in communications engineering from Southwest Jiaotong University, Chengdu, China, in 2006; the M.Sc. degree in electromagnetic fields and wavelength technology from Beijing University of Posts and Telecommunications, Beijing, China, in 2009; and the Ph.D. degree in computer science from University of Nottingham, Nottingham, U.K., in 2013.

He is an Associate Professor with the School of Information Science and Technology, Southwest Jiaotong University. His research interests include evolutionary algorithms, multiobjective optimization, network coding, and routing in communication networks. He has authored or co-authored over 20 peer-reviewed journal and conference papers.

Tianrui Li (SM'11) received the B.S., M.S., and Ph.D. degrees in traffic information processing and control from Southwest Jiaotong University, Chengdu, China, in 1992, 1995, and 2002, respectively.

He was a Post-Doctoral Researcher with SCK•CEN, Mol, Belgium, from 2005 to 2006, and a Visiting Professor with Hasselt University, Hasselt, Belgium, in 2008; University of Technology, Sydney, Australia, in 2009; and University of Regina, Regina, Canada, in 2014. He is currently a Professor and the Director of the Key Laboratory of Cloud Computing and Intelligent Techniques, Southwest Jiaotong University. He has authored or co-authored over 150 research papers in refereed journals and conferences. His research interests include big data, cloud computing, data mining, granular computing, and rough sets.

Dr. Li is a fellow of the International Rough Set Society.

Yan Yang received the B.Sc. degree in electrical engineering and the M.Sc. degree in electrical and communication systems from Huazhong University of Science and Technology, Wuhan, China, in 1984 and 1987, respectively, and the Ph.D. degree in traffic information engineering and control from Southwest Jiaotong University, Chengdu, China, in 2007.

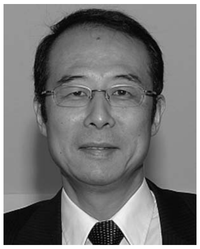
She is a Professor and the Vice Dean of the School of Information Science and Technology, Southwest Jiaotong University. From 2002 to 2003 and from 2004 to 2005, she was a Visiting Scholar with University of Waterloo, Waterloo, ON, Canada. She has authored or co-authored over 100 papers in journals and international conference proceedings, one proceeding, one special issues of international journal, and two books. Her research interests include computational intelligence, data mining, ensemble learning, and cloud computing.

Prof. Yang is the Vice Chair of the ACM Chengdu Chapter, an Executive Member of the China Computer Federation (CCF) Chengdu Member Center, and a Senior Member of CCF and Chinese Association for Artificial Intelligence.

Rong Qu (SM'12) received the B.Sc. degree in computer science and its applications from Xidian University, Xi'an, China, in 1996 and the Ph.D. degree in computer science from University of Nottingham, Nottingham, U.K., in 2003.

She is an Associate Professor with the School of Computer Science, University of Nottingham. She has published over 50 peer-refereed papers in international journals since 2000. Her research interests include the modeling and optimization algorithms in scheduling and optimization algorithms for logistics transport scheduling, personnel scheduling, telecommunication network routing, portfolio optimization, and timetabling problems, by using evolutionary algorithms, mathematical programming, constraint programming in operational research and artificial intelligence, and hybridization of these techniques.

Dr. Qu was the Program Chair of symposiums, workshops, and special sessions with the international conferences including the Symposium Series on Computational Intelligence and Congress on Evolutionary Computation. She is the Chair of Task Force on hyper-heuristics at the IEEE Computational Intelligence Society. She is the Leading Guest Editor of the Special Issues of Artificial Intelligence Planning and Scheduling for *Journal of Scheduling*.



Yi Pan (SM'91) received the B.S. and M.S. degrees in computer engineering from Tsinghua University, Beijing, China, in 1982 and 1984, respectively, and the Ph.D. degree in computer science from University of Pittsburgh, Pittsburgh, PA, USA, in 1991.

He is a Distinguished University Professor and an Interim Associate Dean with the Department of Computer Science, Georgia State University, Atlanta, GA, USA. He is also a Changjiang Scholar Chair Professor with Central South University, Changsha, China, and an IV-Endowed Visiting Chair Professor with Tsinghua University, and a Guest Professor with Beijing University, Beijing. His profile has been featured as a distinguished alumnus in both Tsinghua Alumni Newsletter and the University of Pittsburgh CS Alumni Newsletter. He has published over 150 journal papers with over 50 papers published in various IEEE journals and over 150 papers in refereed conferences. He has also co-authored or co-edited 39 books and there are over 3600 citations that refer to his published work. His research interests include parallel and cloud computing, wireless networks, and bioinformatics.

Dr. Pan was a recipient of several awards, including the IEEE Transactions Best Paper Award, the IBM Faculty Award, the Japan Society for the Promotion of Science Senior Fellowship, the IEEE International Conference on Binformatics and Bioengineering Outstanding Achievement Award, the National Natural Science Foundation Research Opportunity Award, and the Air Force Office of Scientific Research Summer Faculty Research Fellowship. He has served as an Editor-in-Chief or an Editorial Board Member for 15 journals including six IEEE TRANSACTIONS. He has organized several international conferences and delivered over 20 keynote speeches at various international conferences.