Evaluating Reliability/Survivability of Capacitated Wireless Networks

Ozgur Kabadurmus, *Member, IEEE*, and Alice E. Smith, *Fellow, IEEE*

***Abstract*—In telecommunication network design problems, survivability and reliability are often used to evaluate quality of service while usually ignoring link capacity. In this paper, a new metric that combines network reliability with network resilience is presented to measure reliability/survivability effectively for capacitated networks. Capacitated resilience is compared with well-known network reliability/survivability metrics (k-terminal reliability, all-terminal reliability, traffic efficiency, and k-connectivity), and its benefits and computational efficiency are discussed. An application is shown using heterogeneous wireless networks (HetNets). With the growing use of new telecommunication technologies such as 4G and wireless hotspots, HetNets are gaining more attention. The source of heterogeneity of a HetNet can either be the differences in nodes (such as transmission ranges, failure rates, and energy levels) or the differences in services offered in the network (such as GSM and WiFi).**

***Index Terms*—Optimization, resilience, telecommunication network reliability, telecommunication network topology, wireless networks.**

Nomenclature

Acronyms and Abbreviations

AP	Access point.
CR	Capacitated resilience.
DSL	Digital subscriber line.
ES	Evolutionary strategies.
HetNets	Heterogeneous wireless networks.
LAN	Local area network.
MANET	Mobile ad hoc network.
QoS	Quality of service.
RF	Resilience factor.
RP	Relay point.
TE	Traffic efficiency.
WLAN	Wireless local area network.
WMAN	Wireless metropolitan area network.
WMN	Wireless mesh network.

Manuscript received December 3, 2016; revised April 10, 2017; accepted June 1, 2017. Date of publication June 30, 2017; date of current version March 1, 2018. Associate Editor: S. Li. *(Corresponding author: Alice E. Smith.)*

O. Kabadurmus is with the Department of International Logistics Management, Yasar University, 35100, Izmir, Turkey (e-mail: ozgur.kabadurmus@yasar.edu.tr).

A. E. Smith is with the Department of Industrial and Systems Engineering and the Department of Computer Science and Software Engineering, Auburn University, Auburn, AL 36849 USA (e-mail: smithae@auburn.edu).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TR.2017.2712667

Notation

C_{ij}	Cut set combination j of subgroup i.
$G(V, E)$	A graph where V and E represent the sets of vertices and edges, respectively.
n_{UR}	Number of unconnected RPs (to an AP) due to limited capacity or being out of range.
n_{UU}	Number of unconnected users in a network.
p_{ij}	Reliability between user (or device) i and device j.
p_i	Assigned path of user i.
P	Path.
R	Reliability.
S_i	Subgroup i.
U_i	User i.
w_i	Weight of user i.

I. Introduction

RELIABILITY/SURVIVABILITY of telecommunication networks is a popular area of optimization. With the growing use of wireless services, e.g., 3G/4G and wireless hotspots, the topic has assumed more importance. The design of the network is very important for service quality. In general, a telecommunication network design problem is to minimize the cost of a network while ensuring QoS. Although networks vary, the requirement is the same: coverage and reliability/survivability at a low cost. This paper addresses "Resilience" in the network design. Although resilience definitions vary, the most common definition of resilience is the ability of a system returning to its normal state after a disruption [1]. In [2], resilience is defined as the ability to reduce the negative effects of a disruptive event, recover from it, and adapt to it. Therefore, a resilient system is not only reliable but also robust and able to restore its normal operational conditions [3]. The application of this paper is capacitated HetNets with mesh-type structures, but the methodology presented in this paper can be applied to other networks such as wired networks or wireless sensor networks.

A WMN consists of interconnected APs, RPs, and gateways, in which clients (users) connect to APs to access the Internet. In this setting, gateways act as bridges between the wireless infrastructure and the Internet, while RPs relay the traffic [4]. According to [5], RPs and APs are often fixed and electrically powered, and APs are connected to a wired link, such as a LAN, a DSL, or a fiber. Wireless network devices in a mesh network are hierarchical. An illustration of the structure of a mesh network is given in Fig. 1. In this structure, users connect

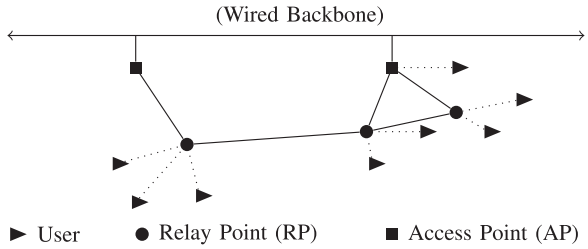

Fig. 1. A WMN design (from [5]).

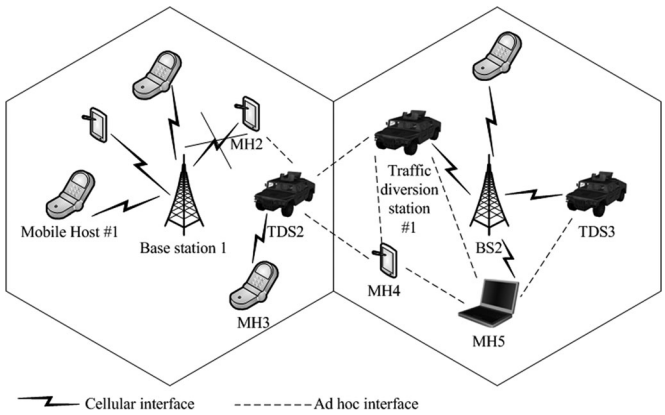

Fig. 2. A HetNet design (from [8]).

to an RP, and RPs connect to an AP, which is connected to the wired backbone.

HetNets integrate various wireless networks and devices. Telecommunication service providers, such as Verizon, Sprint, and T-Mobile, are integrating or planning to integrate multiple wireless technologies with partially overlapped coverage areas [6]. An example is to offer wireless LAN access for their 3G/4G customers.

However, to benefit from these services, users (i.e., mobile hosts) must be equipped with one or more wireless access technologies. In this case, a mobile host can choose WiFi in one location and 3G/4G in another location because of different cost rates, bandwidth, or coverage properties. Because customer satisfaction is closely related to QoS and better service usually means more investment, a service provider will optimize cost by using the most efficient combination of available heterogeneous wireless technologies [6]. The future of wireless networks is to provide seamless mobility to users with the integration of different wireless access technologies [7]. For example, Niyato and Hossain [7] foresee the integration of 802.16-based WMANs and 802.11-based WLANs in the near future. This integration is an example of a HetNet.

Although HetNets with a mesh-type structure is investigated in this paper, there are other applications of HetNets in wireless networks. For example, Yang *et al.* [8] introduce a MANET technology on top of a cellular system (see Fig. 2) to increase system performance. Unlike mesh-type networks where the network has a hierarchy, MANETs are self-configuring networks without fixed infrastructure. The traffic diversion stations and mobile hosts can use both ad hoc technologies and cellular network technologies. In MANETs, dynamic node behavior may affect failure propagation and worsen the status of an impaired system [9]. However, node mobility or epidemic spreading of failures in complex dynamical networks [10], [11] are not considered in this paper.

The competitive structure of the telecommunication industry motivates service providers to invest more in infrastructure to satisfy user demand. Coverage, reliability, and survivability are the main concerns of users. In the network reliability/survivability literature, many metrics have been proposed. Among them, all-terminal reliability, k-terminal reliability, TE, k edge-disjoint paths, and k node-disjoint paths are commonly used. In this paper, a new metric is proposed to address limitations of previous metrics as applied to capacitated networks and to provide a new consideration of survivability.

Connectivity-based metrics (k edge-disjoint or node-disjoint paths) consider neither reliability nor capacity in the network. They focus on redundant paths without considering reliability of nodes nor edges. Terminal reliability (k-terminal or all-terminal) focuses on reliability but does not consider capacity [12]. TE [13] considers rerouting options, but it does not consider capacity.

The new metric herein, Capacitated Resilience (CR), considers capacity, reliability, and rerouting *simultaneously*. CR uses reliability and scales it with rerouting options to find the true resilience of a network under capacity constraints. In [14], resilience is defined as "the intrinsic ability of a system to maintain or regain a dynamically stable state, which allows it to continue operations after a major mishap and/or in the presence of a continuous stress." According to [15], adaptive capacity is another important aspect for resilience. Hence, CR measures resilience by considering rerouting options that permit normal operation in the case of a failure of a designated path. In the case of a disruption in the assigned path, neglecting handoff time, recovery is immediate using rerouting. Therefore, the availability of rerouting options enables a resilient system. CR ranges from 0 to 1, which allows a direct comparison of different network designs unlike connectivity-based metrics. Possibly, in some interdependent systems, capacity of nodes and load redistribution may lead to cascading failures [16]–[18]. However, in this paper, cascading failures and load redistribution are not considered.

The primary objectives of this paper are to present a new and useful survivability metric, capacitated resilience, for HetNets. The main hypothesis of this research is that a network design with better and more practical allocation of redundancies for rerouting options can be obtained by using CR instead of using traditional reliability metrics or k-connectivity (k node-disjoint/edge-disjoint paths) constraints. Network designs optimized by CR are compared with those optimized by traditional reliability metrics to assess this hypothesis.

The following section presents the problem definition and the proposed CR metric.

II. Background

A telecommunication network can be defined as a graph G consisting of vertices (or nodes) of users and network devices.

Edges are the connections among users (or devices) and network devices. To be connected with a network device v, a user or network device must be within the communication range of v. The communication range is determined by the technical specification of the network device. A user is assumed to be connected if reached by a device. Also, the user-device link capacity is assumed to be sufficiently large. Throughout this paper, the sets of vertices (devices, users) and edges (wireless or wired links) of G will be denoted as V and E, respectively.

In wireless telecommunication network design problems, users connect to an intermediate node; then, intermediate nodes connect to an end node (e.g., a base station) usually connected to a wired backbone. The goal is coverage and reliability/survivability at a low cost.

The design of survivable/reliable HetNets is an emerging area of optimization, which has applications in mesh and sensor networks. With the growing use of new telecommunications technologies such as 4G and wireless hotspots, this subject is attracting more attention. The sources of heterogeneity of a HetNet are the difference in services offered in the network (such as 3G/4G and WiFi) and/or the difference in nodes (such as transmission ranges, failure rates, and energy levels). In this paper, the problem includes varied nodes and services within the wireless network.

The impact of wireless interference can be much reduced when multiple wireless network interface cards are used in routers and APs [5]. Interference is at its maximum impact if there is only one wireless channel available; on the contrary, it can be neglected if enough channels and radio interfaces are available in the mesh nodes [5]. By the definition of HetNets, many channels and interfaces (due to the heterogeneity of the network components) operate in the network, and therefore, interference is neglected in our paper. Also, link scheduling methods such as carrier-sense multiple access and time-division multiple access can provide a conflict-free transmission schedule within a WMN [19]–[21]. Also, some operational issues, such as minimum transmission time [22], [23] in stochastic flow networks or channel fading [24] in wireless networks, are not considered in this paper since the main focus is on survivability/reliability.

In this paper, the network structure will be similar to the mesh network structures of [5] and [25]. Although CR can be applied to any type of network, the focus herein is on WMNs. In [25], users connect to a gateway and gateways connect to a sink node. In [5], using a homogeneous wireless network, a structure for mesh networks (see Fig. 1) is presented and the network design problem is solved to minimize the installation cost of the network under a full coverage constraint. The network herein is a mesh [5] with heterogeneous properties [25].

Most of the studies consider survivability/reliability as a constraint (node-disjoint or edge-disjoint paths). For example, Kashyap *et al.* [26] minimize the number of relay nodes in a sensor network under k-connectivity constraints. Benyamina *et al.* [4] present a reliable mesh network design problem with k-connectivity constraints. The remaining papers on optimization of wireless networks do not consider survivability/reliability. Among them, Amaldi *et al.* [5] propose a mathematical model to minimize the cost of a mesh network without considering survivability/reliability. Benyamina *et al.* [27] uses the problem studied by Amaldi *et al.* [5]; however, they minimize cost and maximize network throughput without survivability/reliability constraints. In another similar work, Benyamina *et al.* [28] propose a multi-objective algorithm for the minimum gateway placement problem in mesh networks (without considering survivability/reliability) to minimize cost and congestion of gateways. The models in [27] and [28] only consider different nodes in the networks as the source of heterogeneity. In this paper, the integration of different wireless technologies is also considered, but more importantly, capacity is included in the context of survivability/reliability.

III. Problem Definition

A. Motivation

In the network design literature, requiring k-connectivity constraints, i.e., k node-disjoint or edge-disjoint paths, is the most common way to improve survivability. Other reliability metrics, namely k-terminal reliability [29], all-terminal reliability [30], and TE [13], [31], have also been used. Among them, two-terminal and all-terminal reliability measures are the most frequently found. TE is defined as "the expected percentage of the total traffic that a network can successfully deliver" [13].

The main motivation for developing a new metric is to include rerouting options as well as reliability when considering capacitated links and devices. Connectivity-based survivability constraints and previous reliability/survivability metrics do not consider capacity. However, in practice, if a link (or a device) does not have enough capacity, then traffic cannot be routed on that link (or device).

Traditional reliability measures (two-terminal/all-terminal reliability and TE) do not consider rerouting options. Survivability constraints, such as k node-disjoint and edge-disjoint paths, consider rerouting options; however, they allocate redundant paths to the network without consideration of reliability or capacity. The redundancy provided by k-connectivity increases the chance that a user remains connected to the network, but it causes more slack capacity and, therefore, potentially higher costs. On the other hand, CR prioritizes redundancy by considering rerouting and leads to designs with less, but effectively distributed, slack capacities.

Another important difference from the other metrics is that CR allows split flows in rerouting. That is, traffic can be rerouted on multiple paths. This is a realistic relaxation that has not been considered before in the reliable network design.

Unlike connectivity constraints, CR allows ready comparison of alternative network designs because it is scaled between 0 and 1. Different k-connected network designs can have different CR values.

Table I summarizes the differences among CR and the other reliability/survivability metrics.

Note that "performability," a widely studied metric in the computer and telecommunications network design literature (originally proposed by [32]) combining reliability and performance, is not included in this study. Performability is used

TABLE I
COMPARISON OF RELIABILITY/SURVIVABILITY METRICS

	Two-terminal reliability	All-terminal reliability	Traffic efficiency	k node-disjoint paths	k edge-disjoint paths	Capacitated resilience
Rerouting				X	X	X
Capacity						X
Node Failures			X			
Link Failures	X	X	X			X
Split flows						X

when some components of a system are degradable and the system operates in a degraded mode if one or more components are failed or degraded [33]. Performance of such a system can be measured using "transmission rates," "processing delays," or other QoS metrics [32]. However, its QoS can be measured by reliability metrics [34]. Since the performance of network components is considered to be nondegradable in this paper and users are either connected or not, performability measures are not considered. Similarly, condition assessment for the performance of a degraded unit [35] is not considered in this paper because the network components are assumed to have only up and down states. In the related area of the stochastic flow network design (e.g., in [22], [36]–[40], and [41]), reliability is defined as the probability that the maximum flow is not less than a given threshold. However, in this paper, all user demand must be satisfied by the network.

B. Proposed Metric: CR

The user-level CR, $\mathrm{CR}(U_i)$, is defined in (1). $R(p_i)$ denotes the reliability of the assigned path of user i, where the user (U_i) is assigned to the device whose path has maximum reliability (from the user to an AP). Herein, without loss of generality, nodes are assumed to be perfectly reliable. The resilience factor (RF) scales the reliability of the assigned path. First, all feasible alternative paths (that is, those having available capacity) from user i to all available APs are identified. Then, the RF is calculated by finding the reliability of the alternative path system. In this paper, capacity is defined in terms of device capacities, and edge capacities are found by the minimum capacity of two devices, i.e., the capacity of the edge (i, j) is equal to the minimum of capacities of devices i and j that the edge connects. Therefore, edge capacity constraints ensure that the capacity of nodes are not exceeded.

$$\mathrm{CR}(U_i) = R(p_i) * \mathrm{RF}(U_i),\ i \in \text{Users} \tag{1}$$

The network-level CR is the weighted average of user resiliences in terms of their traffic requirements. If the network consists of one user, then user resilience is equal to network level resilience. Equation (2) presents the calculation of capacitated network resilience, where CR denotes network-level resilience and $\mathrm{CR}(U_i)$ denotes the resilience of user i (U_i). w_i is the weight of user i, which is the proportion of traffic flow of user i to the total traffic flow of all users. Note that the traffic flow of a user is assumed to be constant over time.

$$\mathrm{CR} = \sum_{i \in \text{ Users}} w_i * \mathrm{CR}(U_i) \tag{2}$$

The next sections explain the calculation steps of CR at the user level in detail:

1) Find the most reliable path between the user and an AP.
2) Identify alternative paths between the user and any AP.
3) Determine disjoint subgroups of the alternative paths.
4) Calculate the reliability of the disjoint subgroups.
5) Calculate the reliability of the alternative paths using subgroup reliabilities.

1) Finding the Most Reliable Path Between the User and an AP: There might be more than one AP in a HetNet. For a given user, finding the most reliable path to connect to an AP is important for service quality. The most reliable path from a user to an AP is found by Dijsktra's shortest path algorithm. Dijkstra's shortest path algorithm minimizes the total distance of a path (sum of edge distances) between two nodes. Note that the computational complexity of Dijsktra's algorithm is $\mathrm{O}(|E| + |N| \log N)$ when a Fibonacci heap is used, where E denotes the number of edges. As shown in (3) and (4), instead of minimizing the total distance in Dijkstra's algorithm, the negatives of the logarithms of the edge reliabilities are minimized to maximize $R(P)$, i.e., the reliability of path P between the user and an AP.

$$\log R(P) = \log \prod_{(i,j) \in P} R_{ij} = \sum_{(i,j) \in P} \log R_{ij} \tag{3}$$

$R(P)$ is maximized when $\sum_{(i,j) \in P} \log(1/R_{ij})$ is minimized, because $0 \leq R_{ij} \leq 1$, as given in (4). Therefore, $\log(1/R_{ij})$ is used as edge distance in Dijkstra's shortest path algorithm to maximize reliability.

$$\max \left\{ R(P) = \prod_{(i,j) \in P} R_{ij} \right\} \equiv \min \left\{ \sum_{(i,j) \in P} \log(1/R_{ij}) \right\} \tag{4}$$

After evaluating reliabilities from the user to all available APs (those having enough capacity), the AP with the most reliable path is assigned to the user.

2) Identifying Alternative Paths Between the User and Any AP: After assigning the most reliable AP to a user, the next step is to identify all alternative paths from the user to all available APs. This subproblem is computationally the most expensive part of the CR calculation.

For each available AP, a k shortest path problem is solved to find the k most reliable paths from the user to that AP. Identifying all available paths might be intractable for large networks, therefore limiting the number of paths (k) reduces the computational complexity of this subproblem. The solution is exact if k is sufficiently large.

The k shortest path problem is not new, and there are many algorithms to solve it. Among them, [42]–[45] are the most important ones. The algorithm of [45] is a generalization of the one of [42]. Eppstein's algorithm [43] is faster than Yen's algorithm [42], but it allows repeated vertices (which makes the search space larger). In this paper, Yen's [42] k shortest

Algorithm 1: Pseudocode of Yen's k shortest path algorithm.

for $k = 1$ **do**
  Step 1: Use Dijsktra's method to find the shortest path from a fixed node to other nodes. The result will be A_1.
  Step 1a: Store A_1 into List A.
end for
for $k = 2$ to K **do**
  Step 2: Check if a node sequence $(1) - \ldots - i$ of A^{k-1} is the same as the first i nodes of a previously generated path $j = 1, 2, \ldots, k-1$. Go to Step 3.
  Step 3: Find the shortest path from i to N without including any nodes from $(1) - \ldots - i$ of A^j (which is called R_i^k). Therefore, the shortest path of the spur of A_i^k, which is S_i^k, is found.
  Step 3a: Add A_i^k (joins R_i^k and S_i^k) to candidate List B. Note that only $K - (k-1)$ many items are needed in List B.
  if Number of paths found at Step 3 + number of paths in List $A > K$ **then**
    Stop. K Shortest paths found. Save the paths to List A.
  else
    Step 4: Move path A^k from List B to List A.
    Step 4a: Leave remaining items in B and k++. Repeat steps 2–4 until obtaining K shortest paths.
  end if
end for
return List A (k shortest paths)

path algorithm is used because it provides an effective and easy implementation and permits only simple paths (no loops or repeated vertices).

The pseudocode of Yen's algorithm to find k shortest paths is given in Algorithm 1 (the interested reader may refer to the original article [42] for more information). The worst-case runtime of Yen's algorithm is $O(kN^3)$, where N is the number of nodes [42]. In this paper, N denotes the number of RPs that can be reached by both the user and a specific AP. Therefore, the average-case performance should be significantly better than $O(kN^3)$.

If k alternative paths with enough capacity to deliver the traffic cannot be found, split paths are considered. A split in a telecommunication network means that the traffic of a user is delivered using two or more distinct paths. The total capacity of split paths must be equal to or more than the required traffic. Here, split paths are only considered when there are not enough k unsplit paths available because splits are harder to manage. Splitting the traffic requires a network protocol that supports splits such as MP-DSR [46] or SMR [47]. Fig. 3 illustrates splitting the traffic of a user. Split paths 1 and 2 have limited capacity that cannot deliver the traffic of the user individually. When combined, their total capacity is equal to or larger than the required traffic of the user, and therefore, they are considered as a single alternative path.

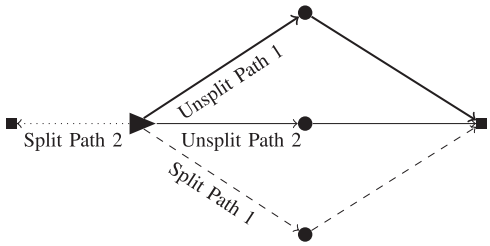

Fig. 3. Splitting the traffic of a user into three paths (Unsplit path 1, unsplit path 2, and split paths 1 and 2 as a path).

Algorithm 2: Pseudocode of clustering alternative paths into disjoint groups.

Save all alternative paths (at most k paths) to List P
groupNo $\leftarrow 1$
for each path $i \in$ P **do**
  if path i is not in a group **then**
    if $i = 1$ **then**
      Assign $groupNo_i \leftarrow groupNo$
      $groupNo++$
    end if
    for $j = 1$ to P **do**
      if i and j has a common edge **then**
        Assign $groupNo_i$ to all paths of the group that j belongs to
      end if
    end for
  end if
end for
return Disjoint subgroups of alternative paths

3) Determining Disjoint Subgroups of the Alternative Paths: Upon identifying all alternative paths from the user to all available APs, the next step is to group the alternative paths into disjoint subgroups. A subgroup consists of a subset of alternative paths of a user. Obviously, a subgroup is a subgraph of the network. For any user having at least one alternative path (except the assigned path), there must be at least one disjoint subgroup of alternative paths. Each disjoint subgroup consists of paths with one or more common edges. Any path of a disjoint subgroup cannot have a common edge with a path of another disjoint subgroup. Disjoint subgroups are determined by a simple algorithm, which is summarized in Algorithm 2. In this algorithm, all alternative paths are compared with each other to check for common edges. If there is a common edge, one of the paths (and all other paths in its group) is labeled as the other path's group. Therefore, the number of disjoint subgroups is dynamic in this algorithm. The worst-case runtime of this algorithm is $O(k^2)$ as it has nested loops.

4) Calculating the Reliability of the Disjoint Subgroups: To find the CR, the reliabilities of the disjoint subgroups (identified in the previous step) must be calculated. However, paths in a disjoint subgroup may not be independent and the reliability of a disjoint subgroup cannot be calculated directly by using

the reliabilities of the paths. Therefore, the minimal cut sets within each disjoint subgroup must be found first to calculate the reliability of that subgroup. In this paper, a minimal cut set is defined as the minimum number of links to disconnect a user(s) from the network, and if any edge of a minimal cut is operational, then the remaining edges in the cut do not disconnect the network [48]. Cut sets do not have to be edge-disjoint.

This problem is similar to the s-t cut set problem, which is basically finding the minimal cut set between source and sink nodes. However, as there can be more than one AP in a subgroup, the problem is not the same as the s-t cut set problem. Another version of this problem is the multiterminal cut problem, which finds a set of edges that disconnect terminal nodes. Xiao [49] and Hartvigsen [50] state that the multiterminal cut problem is NP-hard for $n \geq 3$, where n is the number of nodes in a graph. Again, this problem is not the same as the problem herein because the terminal nodes, i.e., the user and APs, do not necessarily communicate with each other. In other words, APs do not communicate with each other because they only serve as a connection to the wired backbone. Also, unlike MANETs, users are not required to communicate with each other.

Thus, algorithms existing in the literature are not suitable for the CR calculation. Therefore, an algorithm has been developed to find all minimal cut sets to disconnect the user from any AP with which user can communicate directly or indirectly (via RPs). Although estimation methods can be used, such as Monte Carlo simulation (e.g., [51] and [52] for all-terminal reliability estimation, and [13] for TE estimation), they are not faster than the proposed CR calculation, as discussed in detail in Section VI. Specifically, the TE calculation using Monte Carlo simulation is slower than the cut set calculation of this paper (see Table IV). Therefore, an exact approach and an approximation are presented to calculate CR.

The pseudocode for finding minimal cut sets is given in Algorithm 3. This algorithm checks all combinations of edges beginning with one edge, and then with two edges, then three edges, and so on. C_{ij} denotes the jth cut set combination of subgroup i (S_i). If any combination of the edges disconnects the user from all APs in the subgroup, then that combination is a cut. If any c-combination ($c > 1$) of edges uses all edges from a previously found c'-combination ($c' < c$) cut set, that combination is not considered since it is not a minimal cut. The algorithm terminates when there are not enough edges left to form a unique combination or all combinations have been examined. The upper bound on the worst-case runtime of this algorithm is $O(k^c)$, where c denotes the cut set size. However, in practice, it approaches $O(k^2)$ since cut set sizes of 3 or greater are rare.

Upon identifying all minimal cut sets of a subgroup, the reliability of the subgroup is calculated. Failure of any cut set disconnects the user from APs. Therefore, all cut sets must be reliable to make the network reliable. In this paper, the reliability of a subgroup is calculated exactly; however, some studies calculate the reliability for a given minimal cut set using Monte Carlo simulation, as done in [39] for stochastic flow networks. However, since cut sets are not necessarily edge-disjoint, an inclusion–exclusion approach must be used to calculate subgroup reliability. Equation (5) presents the reliability calculation of a subgroup. In this formulation, $R(S_i)$ denotes the reliability of subgroup i and $P(C_i)$ denotes the operational probability of cut set combination C_i.

$$\begin{aligned} \mathrm{R}(S_i) &= 1 - P(C_1 + C_2 + \cdots + C_n) \\ &= 1 - [[P(C_1) + P(C_2) + \cdots + P(C_n)] \\ &\quad - [P(C_1C_2) + P(C_1C_3) + \cdots + P(C_{n-1}C_n)] \\ &\quad + \cdots + (-1)^n [P(C_1 \ldots C_{n-1}C_n)]] \end{aligned} \tag{5}$$

5) Calculating Reliability of the Alternative Paths (Termed the RF) Using Subgroup Reliabilities: After calculating the reliabilities of the disjoint subgroups, the last step is to calculate the reliability of the alternative paths (that is, union of subgroups). This calculation is simply the parallel reliability calculation, where the system is reliable if at least one of the subgroups is reliable. Equation (6) shows this calculation where S denotes the number of disjoint subgroups.

$$\mathrm{RF} = 1 - \prod_{i=1}^{S} [1 - R(S_i)] \tag{6}$$

C. Approximation of CR

CR can be calculated exactly by setting the number of alternative paths and the cut set size large enough. Initial experimentation showed that a cut set size of 4 and number of alternative paths of 10 gave exact CR for test problems of sizes ranging from 10 to 100 users. By setting the number of alternative paths and the cut set size to a smaller number, higher order cut sets are not included and an approximation of CR is obtained. The benefit of approximation is computational savings due to the reduced number of alternative paths and cuts since the upper bound on the worst-case runtime of CR calculation is $O(k^c)$.

Tables II summarizes the approximation error of the CR calculation for a number of users and budget combinations (10 problem instances of each). The results are given in percent gaps of the exact CR value, where a negative gap indicates underestimation of CR and a positive one indicates overestimation. As cut set size decreases from 2 to 1, CR is slightly overestimated due to the limited number of cuts. However, this change is not as significant as the change due to the decreased number of alternative paths. The gap increases as the number of alternative paths decreases because of the reduced number of rerouting options. Also, the gap increases as the budget increases for the same number of users (e.g., budgets of 500, 600, and 1200 of the 10-user problem) because a larger budget allows more devices and therefore more rerouting options. Estimation accuracy increases as the problem size grows, i.e., gaps are lower in the 50 and 100 user problems than in the 10 and 25 user ones. This is a favorable attribute of the method.

A cut set size of 3 gave the exact resilience for all test problems. For cut set sizes larger than 2 to affect the value of CR, at least three edges (that cannot disconnect the network in cut sets of sizes 1 or 2) fail. This scenario requires an unusual case of two or more RPs in a cut set and more than two paths sharing an RP. Moreover, due to the low reliability of such a system, it has

TABLE II
APPROXIMATION ERROR (% GAP OF CR) FOR A NUMBER OF ALTERNATIVE PATHS AND CUT SET SIZE COMBINATIONS COMPARED TO THE EXACT VALUE (OVER 10 PROBLEM INSTANCES EACH)

	Approximation by (Number of alternative paths, cut set size)							
	Alternative paths					Cut set size		
Users, budget	(5, 4)	(4, 4)	(3, 4)	(2, 4)	(1, 4)	(10, 4)	(10, 2)	(10, 1)
10,500	0	0	0	−0.01	−13.86	0	0.02	0.18
10,600	0	0	0	−0.13	−11.66	0	0.02	0.34
10,1200	0	0	0	−0.02	−3.79	0	0.06	0.23
25,1200	0	0	0	−0.05	−10.11	0	0.07	0.18
25,2000	0	0	0	−0.11	−8.01	0	0.28	0.55
50,1700	0	0	0	0	−7.18	0	0.01	0.11
50,2400	0	0	0	−0.05	−4.34	0	0.05	0.22
100,3000	0	0	0	0	−2.56	0	0.01	0.01
100,4000	0	0	0	−0.05	−2.63	0	0.27	0.68
All scenarios	0	0	0	−0.05	−7.13	0	0.09	0.28

Algorithm 3: Pseudocode of finding minimal cut sets of a subgroup.

```
Save all unique edges of alternative paths in the subgroup to set E
Cutsets ← ∅
for c = 1 to (CutSet Size) do
    if |E| - number of unique edges in Cutsets < c then
        break
    end if
    TempCutsets ← ∅
    for each unique c-combination of edges (C_i) ∈ E do
        if C_i ⊉ C_j (C_j ∈ cutsets) then
            if removal of edges in C_i disconnects user from all APs then
                TempCutsets ← TempCutsets + {Edges in C_i}
            end if
        end if
    end for
    Cutsets ← Cutsets + TempCutsets
end for
return Cutsets
```

a very small chance to be selected as one of the k most reliable paths. This result is also noted in the computational complexity discussion of Algorithm 3 in Section III-B4. In terms of the approximation quality for different number of alternative paths, CR can be approximated without losing much accuracy (within 1% of the exact value on average) if the number of alternative paths is set to 5.

The differences in solution time between the exact calculation and the approximations of CR are discussed in Section VI.

IV. ILLUSTRATIVE EXAMPLE

To demonstrate the calculation steps of CR, consider the network presented in Fig. 4. In this network, there are four APs and many rerouting options for a single user.

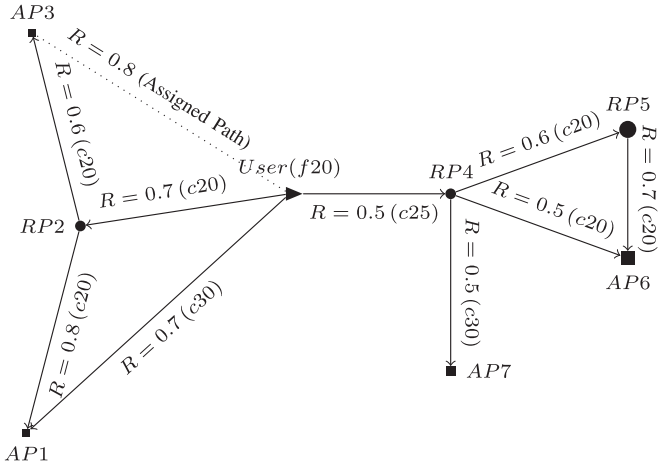

Fig. 4. Example network (R, c, and f denote edge reliability, edge capacity, and user flow requirement, respectively).

To calculate CR of this network, the most reliable path from the user to an AP is identified first (User—AP3). Then, all possible alternative paths from the user to all available APs are found (where capacity is available) by Yen's k shortest path algorithm. For this example, k is sufficiently large ($k = 3$) to find all shortest paths. Then, alternative paths from the user to the APs are grouped into edge-disjoint groups using the procedure given in Section III-B3. Disjoint subgroups 1–3 are shown in Fig. 5(a)–(c), respectively.

The cut set of subgroup 1 [see Fig. 5(a)] is $U - AP1$ and the reliability of the subgroup, $R(S1)$, is 0.7.

The cut set of subgroup 2 [see Fig. 5(b)] consists of $(U - RP2)$ and $(RP2 - AP1,\ RP2 - AP3)$. The reliability of the subgroup is 0.644. The calculation steps are presented below:

$$R_{S2} = 1 - ((1-0.7) + (1-0.6)(1-0.8) - (1-0.7)(1-0.8)(1-0.6)) = 0.644$$

The cut set of subgroup 3 [see Fig. 5(c)] consists of $(U - RP4)$, $(RP4 - AP7, RP4 - RP5, RP4 - AP6)$ and $(RP4 - AP7, RP5 - AP7, RP4 - AP6)$. The reliability of the subgroup is 0.4275. The reliability calculation of subgroup 3 is given below:

$$\begin{aligned} R_{S3} = 1 - ([&(1-0.5) + (1-0.5)(1-0.6)(1-0.5) \\ &+ (1-0.5)(1-0.7)(1-0.5)] \\ &[(1-0.5)(1-0.5)(1-0.6)(1-0.5) \\ &+ (1-0.5)(1-0.5)(1-0.7)(1-0.5) \\ &+ (1-0.5)(1-0.6)(1-0.7)(1-0.5)] \\ &+ [(1-0.5)(1-0.5)(1-0.6)(1-0.7)(1-0.5)]) \\ = 0.4275 \end{aligned}$$

After calculating reliabilities of the subgroups, the reliability of the alternative path system (RF) can be calculated by:

$$\text{RF} = 1 - (1-0.7)(1-0.644)(1-0.4275) = 0.938857$$

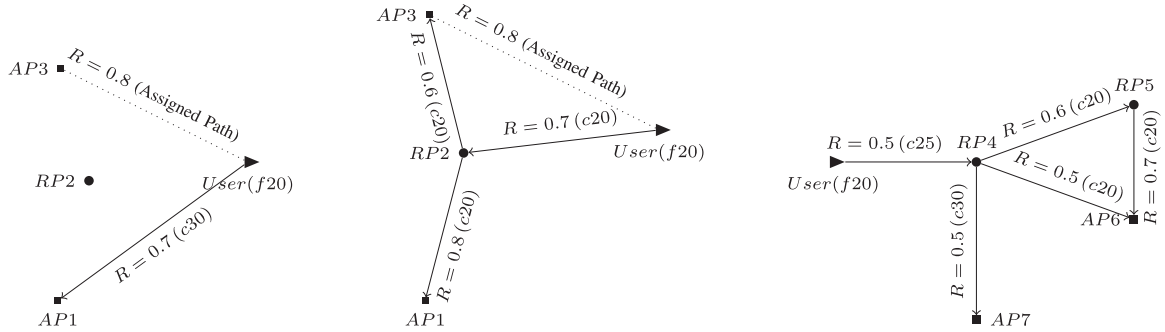

Fig. 5. Disjoint subgroups 1–3 of the network in Fig. 4. (a) Subgroup 1. (b) Subgroup 2. (c) Subgroup 3.

The CR of the network has been defined in (1). Therefore, the CR (for the user) is calculated by:

$$\text{CR} = 0.8(0.938857) = 0.751086$$

Network-level CR is calculated by the weighted average (in terms of flows) of user CRs, and in this example, there is only one user.

V. Network Design Using CR

A telecommunication network can be designed for various objectives, for example, maximum reliability (all-terminal or k-terminal), k node-disjoint, or edge-disjoint paths. However, as proposed in this paper, the CR metric can also be used to design a network.

A network consists of geographically diverse users having different demands. The design problem specifies the locations, types, and numbers of the devices to connect all users to a network in a low-cost and reliable way. This problem is not straightforward to solve, and it includes nonlinearity in the resilience calculations. Therefore, obtaining the optimal solution of the problem is extremely difficult for nontrivial sized networks. However, a metaheuristic can solve this problem effectively.

To design a network by assigning devices to ensure maximum CR (or minimum cost), an ES optimization has been developed. In ES, the population (a set of solutions) improves during generations (iterations) until a termination criterion is met [53]. At each generation, mutation operations are applied to the population to generate new (and often better) solutions. The best of these survive to the next generation. In this paper, ES is used to maximize CR because the optimization problem includes continuous decision variables (device coordinates) and ES is known for its success with such problems. Although the other decision variables are discrete (number of devices, device types, user to device assignments, and routing of traffic flows), the device coordinates are hardest to optimize and ES performs well on the problem presented here. Both single and multi-objective ES algorithms are developed to design resilient HetNets.

Before presenting the ES model, a mathematical formulation of the problem is given in the next section to present the properties of the model.

A. Basic Mathematical Model of the Problem

In this section, the mathematical formulation of the design of resilient HetNets problem is presented. This formulation has been adapted from [5]. In this problem, users ($U = 1, .., n$) are served by devices ($D = 1, ..., m$). The node N represents the wired backbone to which APs connect. Decision variables of the model are

$$x_{ij} = \begin{cases} 1, & \text{if user } i \text{ is assigned to device } j \\ 0, & \text{otherwise} \end{cases}$$

$$a_{ij} = \begin{cases} 1, & \text{if device } j \text{ serves user } i \\ 0, & \text{otherwise} \end{cases}$$

$$b_{jl} = \begin{cases} 1, & \text{if devices } j \text{ and } l \text{ can be connected} \\ 0, & \text{otherwise} \end{cases}$$

$$y_{jl} = \begin{cases} 1, & \text{if devices } j \text{ and } l \text{ are connected} \\ 0, & \text{otherwise} \end{cases}$$

$$z_j = \begin{cases} 1, & \text{if device } j \text{ is in the solution} \\ 0, & \text{otherwise} \end{cases}$$

$$w_{jN} = \begin{cases} 1, & \text{if device } j \text{ is an AP} \\ 0, & \text{if device } j \text{ is an RP} \end{cases}$$

f_{jl} = traffic flow routed on link (j, l);
f_{jN} = traffic flow routed on the wired link between device j and the wired backbone;
$\zeta_{jx} = x$ coordinate of device j;
$\zeta_{jy} = y$ coordinate of device j.

The parameters of the model are
$\zeta_{ix} = x$ coordinate of user i;
$\zeta_{iy} = y$ coordinate of user i;
α_i = flow requirement of user i;
u_{jl} = capacity of link j, l;
u_j = capacity of device j;
r_j = transmission range of device j;
c_{AP} = cost of locating an AP;
c_{RP} = cost of locating an RP;
B = budget for deploying devices;
m = maximum number of devices in the solution;
n = number of users in the network;
M = sufficiently big number.

Using the above decision variables and parameters, the design of resilient HetNets is formulated as follows.

$$\max \text{ CR} \tag{7}$$

s.t.

$$\sum_j x_{ij} = 1 \ \ \forall i \tag{8}$$

$$x_{ij} \leq z_j * a_{ij} \ \ \forall i, j \tag{9}$$

$$\sum_i \alpha_i * x_{ij} + \sum_l (f_{lj} - f_{jl}) - f_{jN} = 0 \ \ \forall j \tag{10}$$

$$f_{lj} + f_{jl} \leq u_{jl} * y_{jl} \ \ \forall j, l \tag{11}$$

$$\sum_i \alpha_i * x_{ij} \leq u_j \ \ \forall j \tag{12}$$

$$f_{jN} \leq M * w_{jN} \ \ \forall j \tag{13}$$

$$y_{jl} \leq z_j, \ \ y_{jl} \leq z_l \ \ \forall j, l \tag{14}$$

$$y_{jl} \leq b_{jl} \ \ \forall j, l \tag{15}$$

$$\sum_j z_j (w_{jN} * c_{\text{AP}} + [1 - w_{jN}] c_{\text{RP}}) \leq B \tag{16}$$

$$x_{ij}, b_{jl}, a_{ij}, y_{jl}, w_{jN}, z_j \in \{0, 1\}, \text{ and}$$
$$f_{jl}, f_{jN}, \zeta_{jx}, \zeta_{jy} \geq 0 \tag{17}$$

The objective function (7) maximizes the CR of the network as explained in detail in Section III-B. Constraint (8) ensures that all users are served by a device. Constraint (9) forces device j to be in the solution if it serves user i. Flow balance constraints are given in (10). Link and device capacity constraints are defined in (11) and (12), respectively. Constraint (13) ensures that the flow is zero between an RP and the wired backbone. Constraints (14) and (15) ensure that the link between devices j and l can be operational only if devices j and l are in the solution and they can communicate with each other. The budget constraint to deploy devices is given in (16). The boundary constraints on the decision variables are given in (17).

Note that radio propagation is beyond the scope of this paper but could be used with this method. Therefore, the reliability of a link is assumed to be inversely proportional to the distance for simplicity. Reliability of the link between devices j and l (p_{jl}) is given in (18). Two devices can communicate only if they are within each other's transmission range. The reliability of the link between user i and device j (p_{ij}) is calculated as given in (19). Device j can serve user i if the user is within the range of the device. Therefore, the decision variable a_{ij} is 1 only if $r_j \geq d_{ij}$. Similarly, the decision variable b_{jl} is 1 only if $\min\{r_j, r_l\} \geq d_{jl}$. All distances are calculated using the Euclidean metric.

$$p_{jl} = \max\left\{0, \frac{\min\{r_j, r_l\} - d_{jl}}{\min\{r_j, r_l\}}\right\} \tag{18}$$

$$p_{ij} = \max\left\{0, \frac{r_j - d_{ij}}{r_j}\right\} \tag{19}$$

To reduce the search space, the maximum number of devices in a solution (m) is limited using the heuristic given in (20). The maximum number can be selected from the range $[B/c_{\text{AP}}, B/c_{\text{RP}}]$, but our extensive experimentation showed that (20) enables a thorough search the feasible region, that is, within the budget. Note that the deployment cost of an RP is lower than an AP, and the ratio may be as high as 1/10 [5]. In this paper, the ratio is assumed to be 1/6.

$$m = \frac{B}{(c_{\text{AP}} + c_{\text{RP}})/2} \tag{20}$$

B. Single-Objective ES

A single-objective ES is developed to design a network with (alternatively) maximum CR, TE, and all and k-terminal reliabilities. This ES model maximizes CR (or the other metrics) under a budget constraint. The budget limits the maximum number of devices in the network as the number of devices and their types change the total cost. The ES selects the numbers and positions of different types of devices (RPs and APs) in the network. It considers capacities of devices and links, and ranges of devices. The pseudocode of the ES is given in Algorithm 4.

In the ES, there are three different mutation types to alter device properties (coordinates and type). The first mutation is to alter device coordinates. In this mutation, the coordinates of a device are changed by normal distributions $N(0, \sigma_x)$ and $N(0, \sigma_y)$ for the x- and y-axes, respectively. The value of σ is dynamically adjusted according to the standard "one-fifth rule" [54]. Mutation success is calculated over a predetermined number of generations (g'). With the second mutation operator, "2-opt swap," two devices are randomly selected and their device types are swapped without changing their coordinates. The last mutation is to change a device type or to remove the device. The coordinate change and swap mutations are performed at each generation. However, the device-type change mutation is performed at every ten generations to give the ES enough time to optimize device coordinates for a given set of coordinates.

The termination criteria of the single-objective ES model are 1000 generations (maxGen) or 250 nonimproving generations (maxNonImprovingGen). A population size (μ) of 30 and children size (λ) of 30 are used, with one child is produced from a randomly selected parent. These parameter values were selected after preliminary experimentation with consideration to the tradeoff between computational time and solution quality. After generating children using mutations, parent and children solutions are pooled and the best solutions are selected for the next generation, i.e., a $(\mu + \lambda)$ replacement policy. The population is randomly initialized. In the initialization procedure, only device types and device coordinates (in continuous space) are created. Cost, reliability, and CR values are calculated accordingly.

Equations (21) and (22) explain the penalty functions to dynamically adjust cost and CR, respectively. The cost of a solution is penalized if a user is not served by any device (21). A user may not be assigned to a device due to lack of device capacity or being beyond the range of any device. The total cost is increased proportional to the number of unserved users (n_{uu}). Another penalty is applied to the cost when the routing of a device is infeasible. Infeasible routing occurs when an RP cannot

Algorithm 4: Pseudocode of the single-objective ES model.

Ensure: Budget constraint, assigning all users to a device, feasible routings
Randomly initialize population (size of $Size_p$)
Sort population (from the best solution to the worst)
bestSolutionSoFar ← Population[0]
$g \leftarrow 0$ //g is the generation counter
while (g < maxGen) **do**
  for ($i = 0$ to $Size_c - 1$) **do**
    Randomly select a parent (i) from population to mutate, $i \in \{0, 1, \ldots, (Size_p - 1)\}$ //$Size_c$ is the children size
    Mutate device coordinates of Child[i]
    Select two random devices j and k of Child[i]
    Swap device types of j and k within Child[i]
    if ($g\%10 = 0$) **then**
      Mutate device types of Child[i]
    end if
  end for
  Sort population and children
  Replace worst population members with newly generated children
  Sort population and update the best solution if necessary
  if ($g\%g' = 0$) **then**
    if (Mutation success rate > 0.20) **then**
      Increase σ_x and σ_y to $1/0.85$ of their values
    else
      Decrease σ_x and σ_y to 0.85 of their values
    end if
  end if
  if (the best solution has not been updated for maxNonImprovingGen generations) **then**
    Terminate ES
  end if
  g ← g+1
end while
return Network design with maximum CR

connect to an AP. The total cost is increased proportional to the number of unconnected RPs to an AP due to limited capacity or being out of range (n_{UR}).

CR is scaled by the percentage of served users (22). It is also scaled by the percentage of devices with feasible paths. Budget is not considered during mutations, but CR is penalized by the cost of the solution. If the total cost of deploying the devices (excluding penalties) is greater than the budget, CR is scaled by "budget/total cost of devices." This dynamically penalizes CR for exceeding the budget and forces the ES to reduce the total number of devices or the number of APs (either by removing them or by replacing them with RPs).

$$\begin{aligned}\text{Cost} = \text{Cost} &+ n_{UU} * \text{penaltyUnservedUser} \\ &+ n_{UR} * \text{penaltyUnconnectedRP}\end{aligned} \tag{21}$$

Algorithm 5: Pseudocode of the bi-objective ES model.

Ensure: Maximum budget constraint, assigning all users to a device, feasible routings
Initialize population, calculate nondominated ranks and crowding distances of solutions
Sort population (according to partial ordering)
bestSolutionSoFar ← Population(0)
g ← 0
while (g < maxGen) **do**
  Mutation operators (same as the single objective ES)
  Sort population and children (according to partial ordering)
  Replace worst population members with newly generated children
  Sort population (according to partial ordering) and update the best solution if necessary
  Update global Pareto front with new solutions
  Add selected Pareto front members to the population
  Adjust σ_x and σ_y (same as for the single objective ES)
  if (bestSolutionSoFar has not been updated for maxNonImprovingGen generations) **then**
    Terminate ES
  end if
  g ← g+1
end while
return The set of Pareto optimal network designs

$$\begin{aligned}\text{CR} = \text{CR} * &\left(\frac{\text{Total number of users} - n_{UU}}{\text{Total number of users}}\right) * \\ &\left(\frac{\text{Total number of devices} - n_{UR}}{\text{Total number of devices}}\right) * \\ &\min\left\{1, \left(\frac{\text{budget}}{\text{total cost}}\right)\right\}\end{aligned} \tag{22}$$

C. Bi-Objective ES

The ES algorithm is extended to a bi-objective optimization to maximize CR and minimize cost using Pareto optimality. A "tradeoff surface" of a set of Pareto optimal solutions (Pareto front) is obtained by Pareto optimality [54], and the decision maker selects from the Pareto front. Therefore, in addition to the population, the bi-objective ES keeps a "global" Pareto front, which includes the best solutions that have been found throughout the ES search. In bi-objective optimization of resilient HetNets, solution i dominates solution j according to the following:

$$\begin{aligned}&\text{Solution } i \text{ dominates } j \text{ if} \\ &\text{Cost}_i \leq \text{Cost}_j \text{ and } \text{CR}_i > \text{CR}_j \text{ , or} \\ &\text{Cost}_i < \text{Cost}_j \text{ and } \text{CR}_i \geq \text{CR}_j\end{aligned} \tag{23}$$

The pseudocode of the bi-objective ES is given in Algorithm 5. "Nondominated rank" and "crowding distance" are adopted from the NSGA-II algorithm [55], a well-known multiobjective

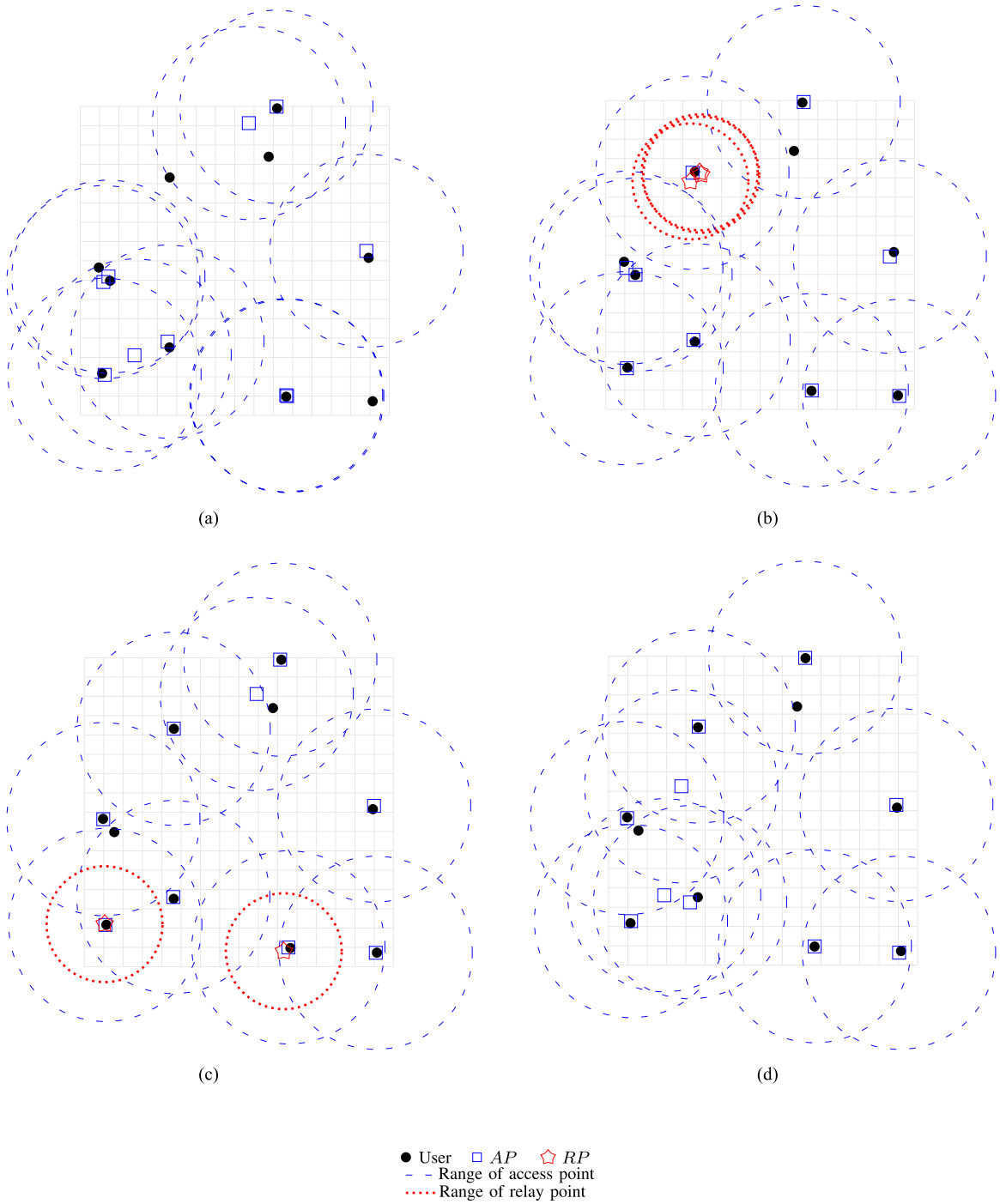

Fig. 6. Summary of the best designs by single-objective ES for a problem instance of ten users, budget = 600, for four metrics: CR, TE, Two-terminal, and All-terminal. (a) Design for CR. (CR: 0.5529, TE: 0.5989, Two-terminal R: 0.7265, All-terminal R: 0.7647, # of APs = 10, # of RPs = 0, Cost = 600). (b) Design for TE. (CR: 0.3267, TE: 0.9859, Two-terminal R: 0.9770, All-terminal R: 0.9787, # of APs = 9, # of RPs = 3, Cost = 570). (c) Design for Two-terminal R. (CR: 0.3695, TE: 0.8565, Two-terminal R: 0.9861, All-terminal R: 0.9862, # of APs = 9, # of RPs = 2, Cost = 560). (d) Design for All-terminal R. (CR: 0.3776, TE: 0.7115, Two-terminal R: 0.9720, All-terminal R: 0.9877, # of APs = 10, # of RPs = 0, Cost = 600).

genetic algorithm, to ensure population diversity in the ES. A solution does not dominate another within a rank. The solutions in the first rank are Pareto optimal because they are not dominated by any solution in the population. Crowding distance, as first defined by Deb *et al.* [55], measures the uniqueness of a solution in objective function space. A higher crowding distance value indicates a more unique solution. It is used to increase population diversity by eliminating similar solutions from the population. Specifically, at the end of each generation, 10% of the least unique solutions (i.e., the solutions with the lowest crowding distance values) are replaced with the most unique solutions (i.e., the solutions with the highest crowding distance values) of the global Pareto front. Solutions are sorted according to "partial ordering" [55]. A solution having a lower nondomi-

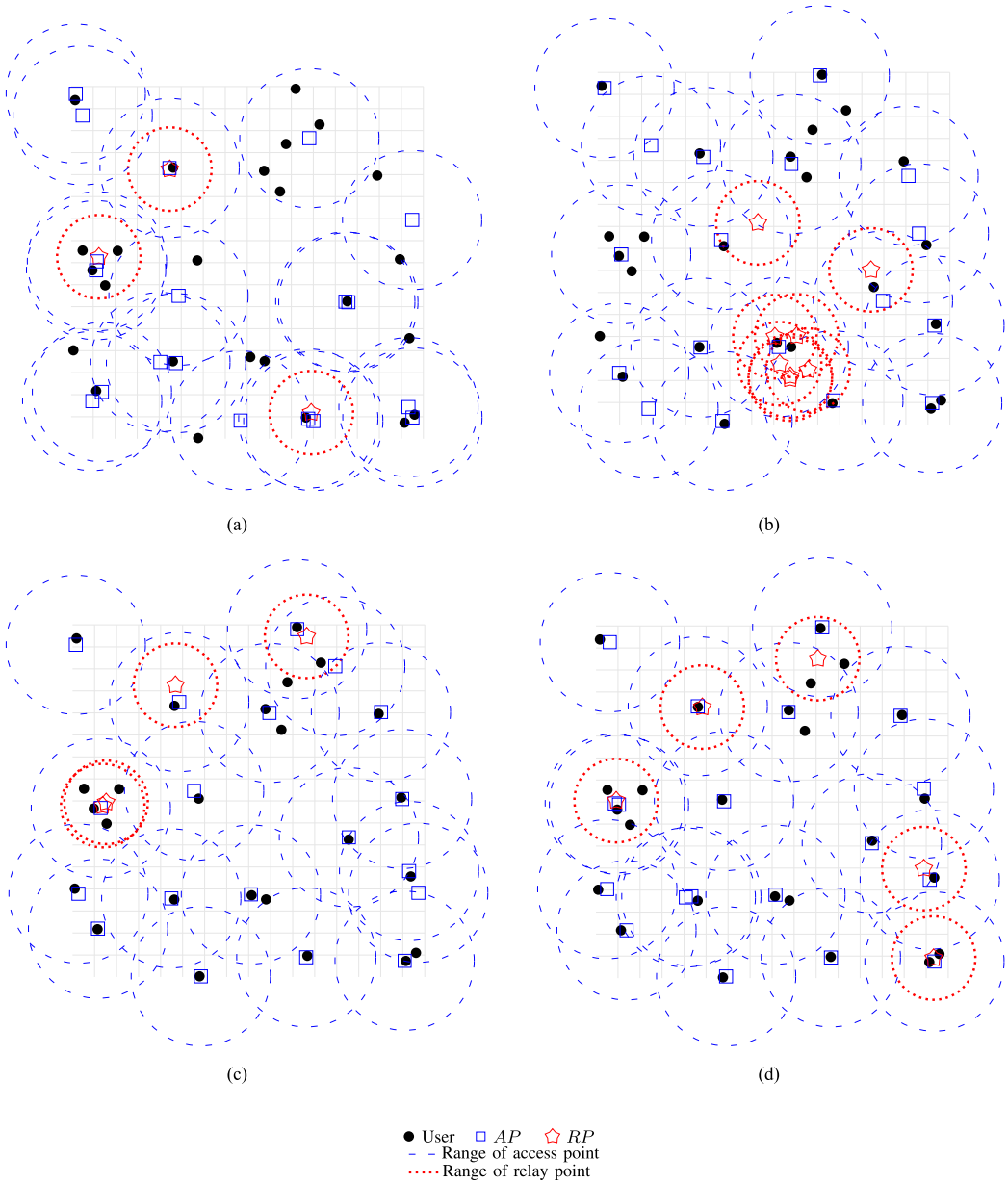

Fig. 7. Summary of the best designs by single-objective ES for a problem instance of 25 users, budget = 1200, for four metrics: CR, TE, Two-terminal, and All-terminal. (a) Design for CR. (CR: 0.4934, TE: 0.4443, Two-terminal R: 0.6638, All-terminal R: 0.7040, # of APs = 19, # of RPs = 3, Cost = 1170). (b) Design for TE. (CR: 0.0706, TE: 0.9084, Two-terminal R: 0.8805, All-terminal R: 0.8848, # of APs = 18, # of RPs = 8, Cost = 1160). (c) Design for Two-Terminal R. (CR: 0.2341, TE: 0.6378, Two-terminal R: 0.9488, All-terminal R: 0.9554, # of APs = 19, # of RPs = 4, Cost = 1180). (d) Design for All-Terminal R. (CR: 0.3294, TE: 0.7674, Two-terminal R: 0.9425, All-terminal R: 0.9619, # of APs = 19, # of RPs = 5, Cost = 1190).

nated rank is better. Within the same rank, solutions with larger crowding distance values are preferred.

The bi-objective ES runs for 2000 generations with an early termination criterion of 500 nonimproved generations and uses the same settings of the single-objective ES for μ, λ and the replacement policy. The selected parameter values were tested over a wide range of test problems, and they perform well for varying problem sizes.

VI. Results

In this section, the differences between the network structures obtained by optimization for CR and the other metrics are compared (see Fig. 6). For this comparison, a 10-user scenario is solved for each metric. Because of the small problem size, differences in the network structures can be visually detected easily.

In the ES, a user may actually represent multiple users in an area. If the number of users in the optimization model is equal to the number of users in the physical world, then that model is a real representation of users. However, this increases the number of users dramatically and, therefore, makes the problem intractable for real-life applications. Hence, the traffic requirement of a user in the model may represent the total traffic requirements of all users in an area. Also, for simplification, a single location in the model may represent multiple nearby natural locations.

TABLE III
SUMMARY OF THE BEST DESIGNS OPTIMIZED FOR ALL METRICS, FOR THE 10-USER AND 25-USER SCENARIOS

Users	Optimized by	Results			
		CR	TE	Two-term	All-term
10	CR	0.5529	0.5989	0.7265	0.7647
	TE	0.3267	0.9859	0.9770	0.9787
	Two-terminal R.	0.3695	0.8565	0.9861	0.9862
	All-terminal R.	0.3776	0.7115	0.9720	0.9877
25	CR	0.4934	0.4443	0.6638	0.7040
	TE	0.0706	0.9084	0.8805	0.8848
	Two-terminal R.	0.2341	0.6378	0.9488	0.9554
	All-terminal R.	0.3294	0.7674	0.9425	0.9619

TABLE IV
SOLUTION TIME COMPARISON OF ALL METRICS FOR THE 10-USER AND 25-USER SCENARIOS (OVER 10 PROBLEM INSTANCES, TEN RANDOM NUMBER SEEDS EACH)

Users	Obj.	Method	Solution Time (s)		
			Average	Min	Max
10	Single	CR	172.70	62.27	899.40
	Single	TE	9557.84	4458.35	13 705.84
	Single	Two-terminal R.	59.26	32.20	94.40
	Single	All-terminal R.	61.49	24.56	99.30
	Bi	Cost and CR	403.81	140.30	763.42
25	Single	CR	545.30	338.21	1907.06
	Single	TE	19 168.93	14 574.02	22 788.97
	Single	Two-terminal R.	229.80	167.12	290.98
	Single	All-terminal R.	241.40	185.84	325.11
	Bi	Cost and CR	1819.02	1107.31	2604.22

Problems with various number of users were randomly generated to assess design differences among the survivability/reliability metrics. Users are randomly located on a square grid of size 4 and 6.325 for the 10-user and the 25-user scenarios, respectively. The grid size increases linearly as the number of users increases to maintain the same density. Traffic requirements of users are randomly assigned between 0 and 20 to incorporate a wide range of user traffic requirements. Transmission ranges of APs and RPs are selected as 2.5 and 1.5 (similar to [5]), respectively. Capacities of APs and RPs are 54 and 20, respectively. User traffic requirements and device capacities are in terms of MB/s. All problem data and results will be made available in online supplement to the published paper.

CR [see Figs. 6(a) and 7(a)] allocates device redundancies in the areas with high traffic requirements. Low traffic flow users are not prioritized due to their smaller weights in the CR calculation. In other words, APs (or RPs) are located near low traffic users only after there is redundancy for high traffic users. Similarly, isolated users connect with distant APs that primarily serve the high traffic users. This is to increase the reliability of the alternative paths of the high traffic users without losing connectivity of low traffic users. High traffic users have connections with a high level of redundancy, while isolated or low traffic users have no or limited redundancy. The allocation of more resources to highly populated areas with larger demand

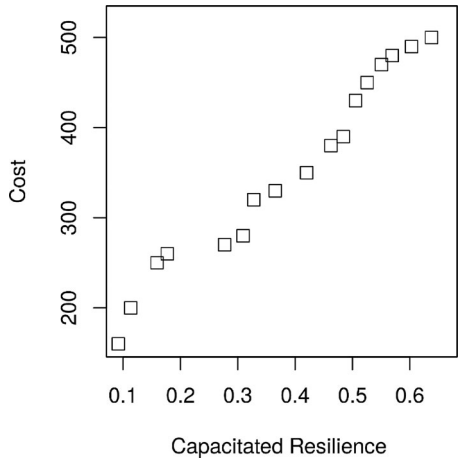

Fig. 8. Sample Pareto front of bi-objective ES (Cost and CR) obtained for the 10-user problem instance.

is a reasonable practice that reflects real life applications. It increases the chance for users to reconnect after a disconnection. This increases the value of CR, but the values of traditional reliability metrics are slightly reduced due to this focused allocation of redundancies. Note that while users are weighted by their demands, other approaches to weighting could easily be used, for example, high weights for critical users such as government and healthcare.

The major difference of the networks found by optimization for TE [see Figs. 6(b) and 7(b)] is the redundancy allocation. There is no or very limited redundancy. Instead, a "dedicated" AP is placed for most of the users. The remaining users share an AP, which is close to other users.

In the two-terminal [see Figs. 6(c) and 7(c)] and all-terminal reliability designs [see Figs. 6(d) and 7(d)], the structures are similar to those obtained by TE optimization. These place a dedicated AP (or RP) for most of the users to achieve higher reliability. However, the design for all-terminal reliability has a higher level of redundancy than the one for two-terminal reliability since it takes all APs into consideration.

Table III compares the designs optimized for all reliability metrics. One problem instance of the 10-user and 25-user scenarios is solved for each metric with ten replications and the best design for each is reported in a row.

As seen from Table IV, the solution time of CR optimization is comparable to those of two-terminal and all-terminal reliabilities. The ES algorithms were coded in Java, and all experiments were carried out on an Intel i7-4790T 2.70 GHz CPU PC with 16-GB memory. TE has the largest solution time due to its computational complexity. (The times differ only by the objective function calculation.) In [13], all solutions are initially run for 300 replications to estimate TE using Monte Carlo simulation, and an additional 5000 replications are performed if the solution is promising with respect to the best found TE value. In this paper, 1000 replications are performed for all solutions. As expected, the solution time of the bi-objective ES (Cost and CR) is higher than that of the single objective. Note that a number of alternative paths of 10 and a cut set size of 4 are used for the CR

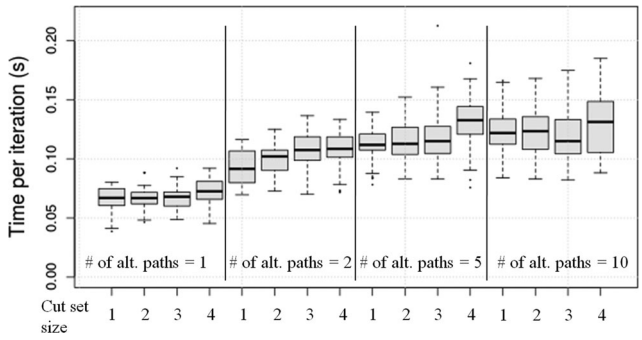

Fig. 9. Solution time per iteration (one single-objective ES generation) comparison according to different number of alternative paths and cut set sizes for CR for the 10-user problem (over 10 problem instances, ten random number seeds each).

calculation in Tables III and IV to obtain a worst-case solution time and exact (or near-exact) value of CR.

The bi-objective ES generates a diversified set of nondominated solutions. A sample Pareto front obtained for the 10-user problem is given in Fig. 8. A decision maker can assess the cost and CR tradeoff and select the best non-dominated solution.

Exact calculation and approximation of CR are also compared. As seen from Fig. 9, solution time per iteration increases as the number of alternative paths increases. However, the time per iteration does not change with the change of the cut set size because the cut set calculation is relatively easy compared to finding alternative paths and the size of the cut sets is usually not more than 2. In almost all test problems, a cut set size of 4 and a number of alternative paths of 5 gave the exact CR.

VII. Discussion

This paper proposes a new metric, capacitated resilience, which considers capacity, reliability, rerouting options, and split flows simultaneously. Capacity and split flows have not been considered by other reliability/survivability metrics in the literature. In telecommunication networks, capacity of devices is a significant constraint for service quality as both devices and links are capacitated. The availability of rerouting options in the case of a failure helps to maintain connectivity and session continuity. Also, splitting the data traffic may prevent congestion and increase the service quality of a network.

In this paper, an exact method to calculate CR is presented as well as an approximation method. The exact calculation is based on the k shortest path calculation and cut set identification. However, a fast and quite precise estimation of CR is obtained by reducing the values of k.

The network design obtained by optimization for CR prioritized high traffic (or highly weighted) users. Redundancies are allocated as the budget allows, but redundant devices are mostly placed near these important users to ensure maximum CR. The other network design metrics prioritize reliable connections for all users (beginning with the high traffic ones) and consider redundancies as a secondary objective. Therefore, it can be concluded that CR better prioritizes the allocation of redundancies (survivability). Redundancies are very important in the case of random failures or planned attacks. From the network survivability perspective, a planned attack has a more severe effect on the network than a random attack (or failure). An attack on the network may aim for removal (or elimination) of some edges (or nodes) to disconnect important users (max-flow min-cut problem). Since the primary design goal of CR is to create redundancies to maintain connectivity in the case of a failure and since CR maximizes survivability by taking cut sets into account, it provides a resilient design that minimizes the adverse effects of planned attacks or random failures.

In this paper, HetNets are used as an application to demonstrate "capacitated resilience," but it could be applied to any network, such as wired telecommunication networks. Although interference has been considered negligible in most of the network reliability/survivability literature, as an extension, the effect of interference could be included in the calculation of CR.

References

[1] S. Hosseini, K. Barker, and J. E. Ramirez-Marquez, "A review of definitions and measures of system resilience," *Rel. Eng. Syst. Safety*, vol. 145, pp. 47–61, 2016.

[2] C. Nan and G. Sansavini, "A quantitative method for assessing resilience of interdependent infrastructures," *Rel. Eng. Syst. Safety*, vol. 157, pp. 35–53, 2017.

[3] Y.-P. Fang, N. Pedroni, and E. Zio, "Resilience-based component importance measures for critical infrastructure network systems," *IEEE Trans. Rel.*, vol. 65, no. 2, pp. 502–512, Jun. 2016.

[4] D. Benyamina, A. Hafid, M. Gendreau, and J. Maureira, "On the design of reliable wireless mesh network infrastructure with QoS constraints," *Comput. Netw.*, vol. 55, no. 8, pp. 1631–1647, 2011.

[5] E. Amaldi, A. Capone, M. Cesana, I. Filippini, and F. Malucelli, "Optimization models and methods for planning wireless mesh networks," *Comput. Netw.*, vol. 52, no. 11, pp. 2159–2171, 2008.

[6] H. Chen, H. Wu, S. Kumar, and N.-F. Tzeng, "Minimum-cost data delivery in heterogeneous wireless networks," *IEEE Trans. Veh. Technol.*, vol. 56, no. 6, pp. 3511–3523, Nov. 2007.

[7] D. Niyato and E. Hossain, "Dynamics of network selection in heterogeneous wireless networks: An evolutionary game approach," *IEEE Trans. Veh. Technol.*, vol. 58, no. 4, pp. 2008–2017, May 2009.

[8] K. Yang, Y. Wu, and H. Chen, "QoS-aware routing in emerging heterogeneous wireless networks," *IEEE Commun. Mag.*, vol. 45, no. 2, pp. 74–80, Feb. 2007.

[9] S. Hong, H. Yang, G. Li, N. Huang, X. Ma, and K. Trivedi, "Analysis of propagation dynamics in complex dynamical network based on disturbance propagation model," *Int. J. Modern Phys. B*, vol. 28, no. 22, 2014, Art. no. 1450149.

[10] S. Hong, H. Yang, E. Zio, and N. Huang, "A novel dynamics model of fault propagation and equilibrium analysis in complex dynamical communication network," *Appl. Math. Comput.*, vol. 247, pp. 1021–1029, 2014.

[11] S. Hong, H. Yang, T. Zhao, and X. Ma, "Epidemic spreading model of complex dynamical network with the heterogeneity of nodes," *Int. J. Syst. Sci.*, vol. 47, no. 11, pp. 2745–2752, 2016.

[12] S. Chakraborty and N. K. Goyal, "Irredundant subset cut enumeration for reliability evaluation of flow networks," *IEEE Trans. Rel.*, vol. 64, no. 4, pp. 1194–1202, Dec. 2015.

[13] A. Konak and M. R. Bartolacci, "Designing survivable resilient networks: A stochastic hybrid genetic algorithm approach," *Omega*, vol. 35, no. 6, pp. 645–658, 2007.

[14] E. Hollnagel, "Resilience the challenge of the unstable," in *Resilience Engineering: Concepts and Precepts*. Aldershot, U.K.: Ashgate, 2006, pp. 9–17.

[15] J. Lundberg and B. J. Johansson, "Systemic resilience model," *Rel. Eng. Syst. Safety*, vol. 141, pp. 22–32, 2015.

[16] S. Hong, B. Wang, X. Ma, J. Wang, and T. Zhao, "Failure cascade in interdependent network with traffic loads," *J. Phys. A: Math. Theor.*, vol. 48, 2015, Art. no. 485101.

[17] S. Hong, X. Zhang, J. Zhu, T. Zhao, and B. Wang, "Suppressing failure cascades in interconnected networks: Considering capacity allocation pattern and load redistribution," *Modern Phys. Lett. B*, vol. 30, no. 5, 2016, Art. no. 1650049.

[18] S. Hong, C. Lv, T. Zhao, B. Wang, J. Wang, and J. Zhu, "Cascading failure analysis and restoration strategy in an interdependent network," *J. Phys. A, Math. Theor.*, vol. 49, no. 19, 2016, Art. no. 195101.

[19] P. H. Pathak and R. Dutta, "A survey of network design problems and joint design approaches in wireless mesh networks," *IEEE Commun. Surv. Tuts.*, vol. 13, no. 3, pp. 396–428, Third Quarter 2011.
[20] L. Xia and B. Shihada, "A Jackson network model and threshold policy for joint optimization of energy and delay in multi-hop wireless networks," *Eur. J. Oper. Res.*, vol. 242, no. 3, pp. 778–787, 2015.
[21] E.-S. Kim and C. A. Glass, "Perfect periodic scheduling for binary tree routing in wireless networks," *Eur. J. Oper. Res.*, vol. 247, no. 2, pp. 389–400, 2015.
[22] W.-C. Yeh, "A fast algorithm for quickest path reliability evaluations in multi-state flow networks," *IEEE Trans. Rel.*, vol. 64, no. 4, pp. 1175–1184, Dec. 2015.
[23] M. El Khadiri and W.-C. Yeh, "An efficient alternative to the exact evaluation of the quickest path flow network reliability problem," *Comput. Oper. Res.*, vol. 76, pp. 22–32, 2016.
[24] L. Xie, P. H. Chong, I. W. Ho, and Y. Guan, "A survey of inter-flow network coding in wireless mesh networks with unicast traffic," *Comput. Netw.*, vol. 91, pp. 738–751, 2015.
[25] A. Capone, M. Cesana, D. D. Donno, and I. Filippini, "Deploying multiple interconnected gateways in heterogeneous wireless sensor networks: An optimization approach," *Comput. Commun.*, vol. 33, no. 10, pp. 1151–1161, 2010.
[26] A. Kashyap, S. Khuller, and M. Shayman, "Relay placement for fault tolerance in wireless networks in higher dimensions," *Comput. Geom.*, vol. 44, no. 4, pp. 206–215, 2010.
[27] D. Benyamina, A. Hafid, and M. Gendreau, "Wireless mesh network planning: A multi-objective optimization approach," in *Proc. IEEE 5th Int. Conf. Broadband Commun., Netw. Syst.*, 2009, pp. 602–609.
[28] D. Benyamina, A. Hafid, and M. Gendreau, "Optimal placement of gateways in multi-hop wireless mesh networks: A clustering-based approach," in *Proc. IEEE 34th Conf. Local Comput. Netw.*, 2009, pp. 625–632.
[29] W. Grover, *Mesh-Based Survivable Networks: Options and Strategies for Optical, MPLS, SONET, and ATM Networking*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2004.
[30] D. Harms, *Network Reliability: Experiments with a Symbolic Algebra Environment* (ser. Discrete Mathematics and Applications). Boca Raton, FL, USA: CRC Press, 1995.
[31] P. Kubat, "Estimation of reliability for communication/computer networks simulation/analytic approach," *IEEE Trans. Commun.*, vol. 37, no. 9, pp. 927–933, Sep. 1989.
[32] J. Meyer, "On evaluating the performability of degradable computing systems," *IEEE Trans. Comput.*, vol. C-29, no. 8, pp. 720–731, Aug. 1980.
[33] A. Goyal and A. Tantawi, "Evaluation of performability for degradable computer systems," *IEEE Trans. Comput.*, vol. C-36, no. 6, pp. 738–744, Jun. 1987.
[34] J. Meyer, "Performability evaluation: Where it is and what lies ahead," in *Proc. IEEE Int. Comput. Perform. Dependability Symp.*, 1995, pp. 334–343.
[35] S. Hong, Z. Zhou, E. Zio, and K. Hong, "Condition assessment for the performance degradation of bearing based on a combinatorial feature extraction method," *Digit. Signal Process.*, vol. 27, pp. 159–166, 2014.
[36] C.-C. Jane, J.-S. Lin, and J. Yuan, "Reliability evaluation of a limited-flow network in terms of minimal cutsets," *IEEE Trans. Rel.*, vol. 42, no. 3, pp. 354–361, Sep. 1993.
[37] Y.-K. Lin, "On a multicommodity stochastic-flow network with unreliable nodes subject to budget constraint," *Eur. J. Oper. Res.*, vol. 176, no. 1, pp. 347–360, 2007.
[38] Y.-K. Lin, "Using minimal cuts to evaluate the system reliability of a stochastic-flow network with failures at nodes and arcs," *Rel. Eng. Syst. Safety*, vol. 75, no. 1, pp. 41–46, 2002.
[39] W.-C. Yeh, "A simple MC-based algorithm for evaluating reliability of stochastic-flow network with unreliable nodes," *Rel. Eng. Syst. Safety*, vol. 83, no. 1, pp. 47–55, 2004.
[40] Y.-K. Lin, "Reliability of a stochastic-flow network with unreliable branches & nodes, under budget constraints," *IEEE Trans. Rel.*, vol. 53, no. 3, pp. 381–387, Sep. 2004.
[41] J. Malinowski, "Reliability analysis of a flow network with a series-parallel-reducible structure," *IEEE Trans. Rel.*, vol. 65, no. 2, pp. 851–859, Jun. 2016.
[42] J. Yen, "Finding the k shortest loopless paths in a network," *Manage. Sci.*, vol. 17, no. 11, pp. 712–716, 1971.
[43] D. Eppstein, "Finding the k shortest paths," *SIAM J. Comput.*, vol. 28, no. 2, pp. 652–673, 1998.
[44] N. Katoh, T. Ibaraki, and H. Mine, "An O(kn^2) algorithm for k shortest simple paths in an undirected graph with nonnegative arc length," *Trans. Inst. Electron. Commun. Eng. Jpn., Sec. E*, vol. 61, pp. 971–972, 1978.
[45] N. Katoh, T. Ibaraki, and H. Mine, "An efficient algorithm for k shortest simple paths," *Networks*, vol. 12, no. 4, pp. 411–427, 1982.
[46] R. Leung, J. Liu, E. Poon, A.-L. Chan, and B. Li, "MP-DSR: A QoS-aware multi-path dynamic source routing protocol for wireless ad-hoc networks," in *Proc. IEEE 26th Annu. Conf. Local Comput. Netw.*, 2001, pp. 132–141.
[47] S.-J. Lee and M. Gerla, "Split multipath routing with maximally disjoint paths in ad hoc networks," in *Proc. IEEE Int. Conf. Commun.*, 2001, vol. 10, pp. 3201–3205.
[48] W.-C. Yeh, "New method in searching for all minimal paths for the directed acyclic network reliability problem," *IEEE Trans. Rel.*, vol. 65, no. 3, pp. 1263–1270, Sep. 2016.
[49] M. Xiao, "Simple and improved parameterized algorithms for multiterminal cuts," *Theory Comput. Syst.*, vol. 46, no. 4, pp. 723–736, 2010.
[50] D. Hartvigsen, "The planar multiterminal cut problem," *Discrete Appl. Math.*, vol. 85, no. 3, pp. 203–222, 1998.
[51] B. Dengiz, F. Altiparmak, and A. E. Smith, "Efficient optimization of all-terminal reliable networks, using an evolutionary approach," *IEEE Trans. Rel.*, vol. 46, no. 1, pp. 18–26, Mar. 1997.
[52] D. Deeter and A. E. Smith, "Economic design of reliable networks," *IIE Trans.*, vol. 30, no. 12, pp. 1161–1174, 1998.
[53] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies—A comprehensive introduction," *Natural Comput.*, vol. 1, no. 1, pp. 3–52, 2002.
[54] J. Dréo, A. Pétrowski, P. Siarry, and E. Taillard, *Metaheuristics for Hard Optimization*. Berlin, Germany: Springer, 2006.
[55] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSG A-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

Ozgur Kabadurmus (M'17) received the B.S. and M.S. degrees in industrial engineering from Istanbul Technical University, Istanbul, Turkey, in 2005 and 2008, respectively, and the M.S. and Ph.D. degrees in industrial engineering from Auburn University, Auburn, AL, USA, in 2011 and 2013, respectively.

He is an Assistant Professor with the Department of International Logistics Management, Yasar University, Izmir, Turkey. His main research interests include applied operations research/metaheuristic optimization, the design of telecommunication systems, and the analysis and design of production systems.

Alice E. Smith (F'17) is the Joe W. Forehand/Accenture Distinguished Professor of the Department of Industrial and Systems Engineering, Auburn University, Auburn, AL, USA, where she served as the Department Chair from 1999 to 2011. She also has a joint appointment with the Department of Computer Science and Software Engineering. She holds one U.S. patent and several international patents and has authored more than 200 publications, which have garnered over 2,800 citations and an H Index of 24 (ISI Web of Science). Her research focuses on analysis, modeling, and optimization of complex systems with an emphasis on computation inspired by natural systems.

Dr. Smith is an Area Editor of the *INFORMS Journal on Computing* and *Computers & Operations Research* and an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING. She has been a Principal Investigator on over $7 million of sponsored research with funding by NASA, the U.S. Department of Defense, the Missile Defense Agency, the National Security Agency, the National Institute of Standards and Technology, the U.S. Department of Transportation, Lockheed Martin, Adtranz (now Bombardier Transportation), the Ben Franklin Technology Center of Western Pennsylvania, and the U.S. National Science Foundation, from which she has been awarded 16 grants including a CAREER grant in 1995 and an ADVANCE Leadership grant in 2001. She is a Fellow of the Institute of Industrial Engineers, a senior member of the Society of Women Engineers, a member of Tau Beta Pi and the Institute for Operations Research and Management Science, and a Registered Professional Engineer in Alabama and Pennsylvania. She was elected to serve on the Administrative Committee of the IEEE Computational Intelligence Society from 2013–2018 and as IIE Senior Vice President Publications from 2014 to 2017.