Priority driven Call Scheduling in Mobile Networks: A MOGA based approach

PARAG KUMAR GUHA THAKURTA[1]
SUJOY SETT[2]

NIT Durgapur
India
[1]parag.nitdgp@gmail.com
[2]sujoysett@gmail.com

Abstract. Enhanced mobile communication in current days demands a drastic evolution in resource management strategies to meet the quality of service (QoS) requirements for the network. Call scheduling is such an important scheme for efficient utilization in spite of scarcity of resources. In this paper, a priority driven call scheduling technique is proposed for efficient routing in mobile networks. The proposed scheduler is addressed with two conflicting objectives: to minimize the mean routing cost, and to achieve uniformity in the overall system utilization. In this context, a Multi Objective Genetic algorithm (MOGA) based approach is introduced for finding an optimal route from several alternatives pertaining to the network constraints. This results in effective scheduling and simultaneous improvement in call acceptance for the system. The proposed model takes into account the significance of priority in a comprehensive manner. The performance of the proposed model is evaluated in network scenarios with different parameters and service requirements. In addition, results from simulation studies show the quality of approximation obtained by the proposed model.

Keywords: call scheduling, optimal route, call acceptance, priority, multi objective optimization (MOO), multi objective genetic algorithm (MOGA)

(Received July 16th, 2014 / Accepted February 4th, 2015)

1 Introduction

In current days, resource management strategies are becoming increasingly important to guarantee the quality of service (QoS) in end-to-end communication between mobile devices. Considering limited amount of resources, the performance of the system in terms of call acceptance depends on efficiency of the applied resource management schemes. Call scheduling is such a scheme which is significantly important in mobile networks for efficient resource utilization even during congestion [25]. Simultaneously, uniformity of load distribution upon the entire network should be considered for fairness and improved performance. Hence, an efficiently designed scheduling scheme is expected to achieve both requirements.

The standard cellular layout [23] through a layered framework is used for scheduling. Each cell representing areas of the layout has one Base Station (BS). The BSs assist in communication between users belonging to various cells, which again in turn communicate between themselves via a Mobile Switching Centre (MSC). Thus, communication between source and destination BSs follows a joint routing scheme. For illustration, the communication from source BS to MSC follows a dedicated route, whereas that from MSC to destination BS uses a different one as shown in Figure 1 following the arrowheads.

In the area of emerging communication services, scheduling algorithms [22] suffer from several shortcomings as follows. A route between two BSs can

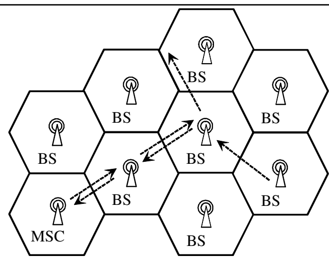

Figure 1: Joint routing scheme

experience good channel conditions in one direction while simultaneously experiencing congestion in the other, thus resulting in delayed scheduling in spite of resource availability. Further, exhaustive search methods to determine the optimal route are computationally hard [17]. Such exact algorithms can only be used for small scale networks and prototype models. Hence, evolutionary algorithms [15] such as genetic algorithm (GA) are used to obtain satisfactory results under this circumstance as it reduces computation overhead. This makes such approaches scalable to larger networks [13].

Multi Objective Optimization (MOO) [18] approach is comprehensively used to address interrelated conflicting issues associated with such scheduling schemes. It generates a set of optimal solutions, known as the Pareto solutions, which lie on different Pareto fronts (F_i) [11]. The traditional methods [15] usually convert the MOO into single objective using a heuristic estimation of the contribution of each objective in terms of their weight values. Here, one objective might dominate another superiorly due to wrong estimation of these weights. In such case, many Pareto optimal solutions are lost and solutions obtained lack diversity. On the other hand, evolutionary algorithms are not sensitive to search space and provide numerous optimal solutions at one time.

The approach known as Multi Objective Genetic Algorithm (MOGA) can be used to handle the MOO problems using the concept of GA. The wide applicability of MOGA over conventional optimization methods has been successfully established. For example, the traveling cost and the total time to traverse in TSP [7] are optimized in such a way that preferential objective is satisfied as both parameters change in different weather or traffic circumstances.

In this paper, a priority based call scheduler through Multi Objective Genetic algorithm (MOGA) based approach is proposed for efficient routing in mobile networks. Before presenting the detailed scope of the work, it might be mentioned here that the terms path and route have been used interchangeably in this work. The BS in each cell is assumed to be a node for implementation. The proposed model is structured in a matrix representation where every node holds a reference to its neighboring nodes in the higher radial level. Several constraints have been pertained on the nodes and their interrelation. Simultaneously, scheduling requests are characterized by their respective priorities. In order to maximize call acceptance of the system through selection of an optimal route, the proposed scheduler is addressed as a MOO problem. It is involved with two objectives: (a) to minimize the mean routing cost, and (b) to achieve uniformity in the overall system utilization. Since these are found to be conflicting with each other, a MOGA based solution is introduced. The initial population of possible alternative routes is identified and the optimal one among those is determined. The standard GA operations are used iteratively until desired solution is obtained. The performance of the proposed model is evaluated in network scenarios with different parameters and service requirements through simulation studies. In addition, it shows the quality of approximation obtained by the proposed model.

The rest of the paper is organized as follows. Section 2 reviews several existing publications related with the proposed scheme for completeness of the work. Section 3 presents the system model and the problem is described in Section 4. Next, the solution of the problem is presented in Section 5. The simulation results are shown in Section 6 followed by conclusions in Section 7.

2 Related Work

Various scheduling techniques have been used to attain the QoS requirements for routing in earlier works. A concept on priority based scheduling [12] used a heuristic estimation to improve the throughput, which was enhanced by introducing a correlation between scheduling cost and users' demand [26]. Another work in [25] brought the concept of priority users enjoying privileges over the conventional users. Tree based routing strategies [9] used different approaches compared to [23, 24] for improving the number of transmissions. A Schedule-based Greedy Expansion (S-Expand) algorithm [8] has been proposed to achieve higher revenue in terms of several network parameters. The techniques used in [20, 27] allow the network operator to prioritize usage of the resources.

High resource demanding services involving end-to-end communication between source and destination mobile users is under the area of current research. An uplink scheduling algorithm [1] in a single cell network used a utility function characterizing QoS of each mobile in terms of throughput. The authors in [2] extended this work under multi-cell network environment. Another joint scheduling algorithm presented in [22] can provide a certain level of coordination among a group of BSs within the wireless cellular networks. It avoids resource wastage and subsequently improves QoS in terms of call acceptance. In [3], a heuristic priority in packet based scheduling algorithm was proposed while providing guaranteed throughput. An increased number of active connections are obtained by an end-to-end scheduling algorithm [16] for efficient utilization of available facilities.

A novel work [4] by Badia et. al. proposed a GA based approach to simultaneously solve link scheduling and routing between a pair of nodes. It evaluates an association between fast computation and the overall system performance, however, not to a desired extent in large scale networks. Another routing scheme [14] investigated the usability of GA operators in influencing the resulting solutions for larger networks. This was further explored in [19] to obtain improved performance in call forwarding topology. Camelo et al. [6] proposed a method for solving routing problems using the Non-dominated Sorting based Genetic Algorithm-II (NSGA-II) [11] for finding different alternatives to guarantee the QoS requirements. Furthermore, Cui et al. [10] studied MOO relying on GAs for determining specific routes.

In short, most of the authors resolved the routing problem in context of call scheduling by optimizing the network performance. However, the overall performance can be further optimized by performing priority driven scheduling. To obtain optimal solution under this circumstance, an exhaustive search grows unboundedly in complexity for larger networks. However, MOGA used in route optimality has attracted attention for its capability of addressing problems with conflicting objectives and supporting parallel computation as well as acquiring multiple non-dominating solutions in a single run. Thus, a MOGA based approach is attempted in this work for obtaining optimal results within reduced time which is particularly important for interactive mobile services.

3 System Model

The alignment of the cellular layout in Figure 2 represents MSC as the central node and the nodes representing BSs surround the MSC in concentric circles. The area under the transmission range [24] of MSC is divided into six hexant slices. Accordingly, each node in this range is associated with a value h [$h \in 0, 1, 2, 3, 4, 5$] which is necessary for establishing the spatial relation between the nodes. The dotted straight lines representing the spokes of the layout are used to partition the nodes, which in turn determines the value of h. With reference to MSC, the nodes in between the spokes and the nodes lying upon the spoke to the right of a slice obtains the corresponding value of h. Any of the six spokes can be selected as reference for the model with a value $h = 0$, and other spokes are incrementally counted in anticlockwise direction from this reference. However, the direction has little impact on the entire problem which is ignored due to symmetric nature of the layout.

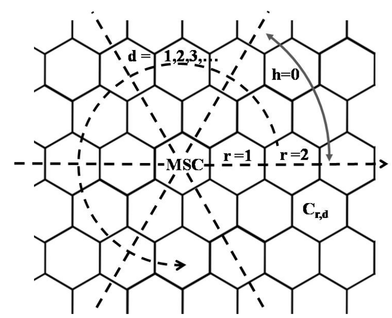

Figure 2: Organization of BSs and MSC in cellular layout

Each node in the layout is represented as $C_{r,d}$, where r and d denote the radial displacement of the node from MSC and the circular displacement of it from the reference spoke respectively. Thus r corresponds to each concentric circle of BSs growing outwardly from the MSC, and d denotes the displacement within the circle from the reference spoke. Here, MSC is denoted by $C_{0,0}$. Any other node $C_{r,d}$ belongs to a candidate hexant which is obtained by $h = \lfloor (d-1)/r \rfloor$ due to anticlockwise assignment in the proposed model. The spoke with the arrowhead (in Figure 2) represents the reference spoke from where the circular displacement d of the nodes is calculated. In addition, the transmission range of MSC is represented by lim_R which is used to determine the maximum number of nodes (N) present in the layout. It is evident from Figure 2 that the number of nodes in each concentric circle of the layout originating from MSC grows in multiples of six. Thus, the number N excluding MSC for a given lim_R is determined by an arithmetic progression which is expressed as follows.

$$N = 1 + (3 \times lim_R \times (lim_R - 1)) \qquad (1)$$

3.1 Node Classification

The nodes $C_{r,d}$ representing BSs are of two types - (a) radial node (R) and (b) non radial node (NR). Radial nodes are those lying on the spokes of the layout. All others are called non radial nodes. These are represented as follows.

$$\text{nodes} = \begin{cases} \text{R} & \text{when } d = (h.r+1) \; \forall \; r \geq 1 \\ \text{NR} & \text{otherwise} \end{cases} \tag{2}$$

Every node has two interaction modes, viz., (a) immediate adjacency (IA) and (b) distant association (DA). IA signifies the direct interaction between two nodes belonging to consecutive radial levels, whereas, DA indicates the same when established through other nodes belonging to the lower radial level. For example (in Figure 3), radial node $C_{2,9}$ shares IA with nodes $C_{3,12}$, $C_{3,13}$ and $C_{3,14}$, and DA with nodes $C_{3,9}$ and $C_{3,17}$. Similarly a non radial node $C_{2,12}$ shares IA with nodes $C_{3,17}$ and $C_{3,18}$, and DA with nodes $C_{3,15}$ and $C_{3,2}$. In addition, the number of nodes sharing IA with a radial node is higher than that of a non radial node, whereas both radial and non radial nodes enjoy equal privileges for DA. The notations RIA and NRIA denote IA for radial and non radial nodes respectively. Similarly, RDA and NRDA are considered for DA.

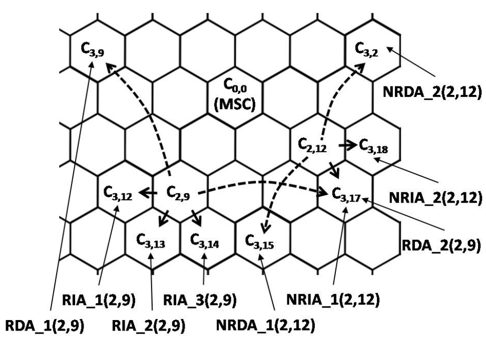

Figure 3: Interaction modes for nodes in cellular layout

3.2 Node Interrelation Matrix (NIM)

The spatial association of the nodes (in Table 1) in the layout is stored as NIM. The adjacency matrix for the entire network is not required by the scheduler as each node maintains interrelation information of the few nodes of higher radial level. This information is essential for finding the candidate routing paths. Thus, NIM results in improved storage space efficiency and reduced scheduling overheads for the entire network.

Table 1: Node Interrelation

	$Radial$	Non_Radial
IA	RIA_1(r,d): if $(d+h-1) \geq 1$, then $C_{r+1,d+h-1}$, else $C_{r+1,6(r+1)}$ RIA_2(r,d): $C_{r+1,d+h}$ RIA_3(r,d): $C_{r+1,d+h+1}$	NRIA_1(r,d): $C_{r+1,d+h}$ NRIA_2(r,d): $C_{r+1,d+h+1}$
DA	RDA_1(r,d): if $(d-h) \leq 0$, then RIA_1(r,5r+1), else RIA_1(r,d-r) RDA_2(r,d): if $(d-r) > 2r$, then RIA_1(r,1), else RIA_1(r,d-r)	NRDA_1(r,d): RIA_1(r,rh+1) NRDA_2(r,d): if $h+1 > 5$, then RIA_3(r,1), else RIA_3(r,(h+1)r+1)

3.3 Inputs

The scheduler is designed to accept a set of call requests characterized by source node, destination node, request priority and permissible tolerance as inputs. The scheduler executes joint routing taking MSC as reference. The scheduler has the capacity of re-scheduling a previously established low-priority request in case another request of higher priority is attempted to be routed through a congested path. Further, the number of hops between source and destination nodes is restricted by the allowable end-to-end tolerance.

3.4 Constraints

The communication through a network is involved with various parameters associated with the BS as well as the links between them which are respectively classified as Node Constraints (NC) and Node Association Constraints (NAC). Parameters such as BS congestion, BS confidence, user density, etc. characterize the former, where as channel capacity, delay heuristics, etc. denote the latter.

The resident traffic in a network determines subsequent call handling capacity. The call volume of each BS must not exceed this capacity. Further, the ratio between the numbers of times a node takes part in routing a call to the number of times it appears in a candidate path while executing the scheduler denotes the BS confidence. The number of users present in a node defines the user density of that node. It provides a look-ahead measure on the expected congestion. In addition, the channel capacity has separate magnitude for joint routing scheme so that the expected delay while availing a channel is different in both directions.

These parameters are conditionally used by the scheduler to determine an optimal route. For illustration, the scheduler forwards a high priority call through a sub optimal route based on nodes with high BS confidence and low user density rather than an optimal route through nodes or links expecting congestion in near future.

4 Problem Description

The scheduler design is associated with the selection of an optimal route for each of the requests (Req_i) in the input set pertaining to the constraints while simultaneously improving the number of calls accepted in the system over a period of time. In this context, the two decision variables involved are - (a) the mean utilization (μ) of the path for routing request Req_i, and, (b) overall uniformity of the system utilization. The capacity of the nodes and links involved in routing determines the mean utilization (μ) of the route. It needs to be minimized for efficient utilization of resources. The standard deviation (σ) on the consumption of each node and link capacity from the mean determines the overall uniformity of the system utilization. Simultaneously, it is to be minimized for distributing the resource consumption throughout the system. Hence, a MOO problem addressing this scenario should obey the following.

$$\text{objectives} \begin{cases} \text{minimize } (\mu) \\ \text{minimize } (\sigma) \end{cases} \tag{3}$$

Here, the variables μ and σ for request Req_i are expressed as follows.

$$\mu_{Req_i} = \frac{1}{l_i} \sum_{\text{Node } C_{r,d} \in \text{route } (Req_i)} cost^i_{C_{r,d}} \tag{4}$$

where, l_i and $cost^i_{C_{r,d}}$ denote route length for request Req_i and routing cost of request Req_i at node $C_{r,d}$ respectively, and

$$\begin{aligned} \sigma_{Req_i} = &\frac{1}{2}(\sigma(\forall \text{ node } C_{r,d} \in \text{route}(Req_i), \\ &\text{node capacity}_{C_{r,d}}) \\ &+ \sigma(\forall \text{ adjacent nodes } C_{r1,d1}, C_{r2,d2} \\ &\in \text{route}(Req_i), \text{link capacity}_{C_{r1,d1},C_{r2,d2}})) \end{aligned} \tag{5}$$

subject to,

$$\begin{aligned} &\forall \text{ node } C_{r,d} \in \text{network} \\ &\qquad \sum Req_i \text{ initiated at } C_{r,d} \leq \text{user density} \\ &\qquad \sum Req_i \text{ destined to } C_{r,d} \leq \text{user density} \end{aligned} \tag{6}$$

$$\begin{aligned} &\forall \text{ scheduled route} \in \text{network}, \\ &\quad \forall \text{ node} \in \text{route}, \\ &\qquad \text{resource consumption} \leq \text{NC} \\ &\quad \forall \text{ node association} \in \text{route}, \\ &\qquad \text{resource consumption} \leq \text{NAC} \end{aligned} \tag{7}$$

In (6), the requests initiating from or getting directed to a node in the network should not exceed the user density of the respective node. In addition, the route for every scheduled request in the network should abide by the node constraints and node association constraints as in (7), which in turn can be expressed in terms of μ_{Req_i} and σ_{Req_i}.

5 Proposed Scheduler

The scheduler executes in sequential phases. In the first phase, the NIM is initialized for a network along with both the constraints and set of call requests. Next, the optimal route, based on the previous information, is obtained through MOGA based approach which is subsequently used for the priority driven scheduling.

5.1 Phase I

Starting from MSC, the interrelation of every node with the nodes of its next higher level is used to generate the NIM following Table 1. This is continued until lim_R is reached. Without loss of generality, the rows (M_i) of NIM obtained from (1) are presented by (8).

$$M_i = \begin{cases} C_{0,0} & \text{where } i = 0 \\ C_{r,d} & \text{where } i = 1 + 3r(r-1) + d \end{cases} \tag{8}$$

The columns of NIM are generated corresponding to the rows, with values denoting the IA and DA of every node. The IA for MSC is populated with initial values, and thereafter IA and DA of the nodes are generated using successive iterations. This procedure is described by algorithm 1.

Based on NIM, the information representing the scheduling requests and associated constraints are randomly generated which are stored in shadow matrices. Two of these matrices having size of $[N \times 6]$ denote the node association constraints, one in direction towards MSC and the other away from MSC. Another column vector known as Node Constraint Matrix $[N \times 1]$ is maintained alongside. An example of corresponding matrices is shown in Figure 4.

5.2 Phase II

An optimal route between the source and destination nodes through MSC for a call request is determined in

Algorithm 1 NIM Generation

```
NIM(1,1)= C_{0,0};
for idx1: 1 to 6 do
    NIM(1,idx1+1)= C_{1,idx1};
for idx1: 1 to lim_R do
    for idx2: 1 to 6*idx1 do
        mIdx=1+3*idx1*(idx1-1)+idx2;
        NIM(mIdx,1)= C_{idx1,idx2};
        if C_{idx1,idx2} ∈ R then
            NIM(mIdx,2)=RIA1(C_{idx1,idx2});
            NIM(mIdx,3)=RIA2(C_{idx1,idx2});
            NIM(mIdx,4)=RIA3(C_{idx1,idx2});
            NIM(mIdx,5)=RDA1(C_{idx1,idx2});
            NIM(mIdx,6)=RDA2(C_{idx1,idx2});
        else if C_{idx1,idx2} ∈ NR then
            NIM(mIdx,2)=NRIA1(C_{idx1,idx2});
            NIM(mIdx,3)=NRIA2(C_{idx1,idx2});
            NIM(mIdx,5)=NRDA1(C_{idx1,idx2});
            NIM(mIdx,6)=NRDA2(C_{idx1,idx2});
return NIM
```

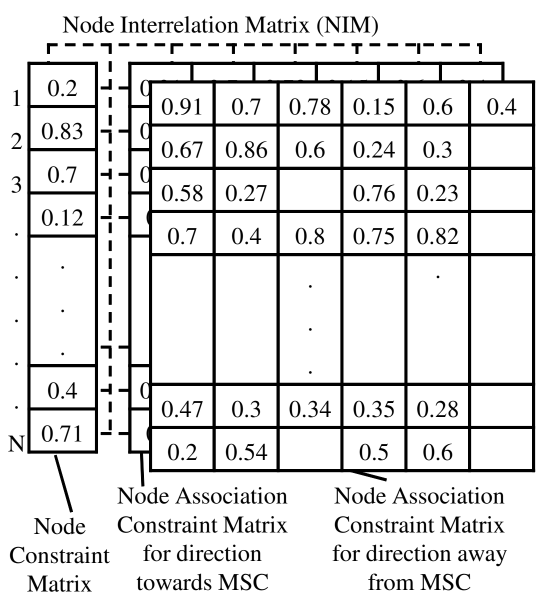

Figure 4: Constraint Matrices

several steps. Two sets of candidate paths, i.e., one from source node to MSC, and another from MSC to destination node are determined from NIM. For example, if a route from MSC to $C_{r,d}$ has to be searched, then $C_{r,d}$ is looked up from rows $((r*(r-1)*3+1)-(6*(r-1)))$ to $(r*(r-1)*3+1)$ of NIM to find the set of nodes in $(r-1)$ that has IA or DA with it. Corresponding to each of these nodes at $(r-1)$, similar look-up is applied to get nodes at $(r-2)$ and this is continued until the look-up reaches at MSC. Then, the optimal candidate from each set is obtained using respective constraints. Finally, these two are merged to provide the desired route between source and destination nodes.

Selection of an optimal path among the alternatives is involved with two objectives as discussed earlier. To understand the interrelation between the objectives, which is essential for ascertaining the nature of the problem, an example is illustrated. Let us assume two candidate paths with random node constraint values $\{0.8, 0.2, 0.1, 0.3, 0.2\}$ and $\{0.6, 0.5, 0.55, 0.6, 0.55\}$ respectively. The μ values for both paths are obtained as 0.32 and 0.57, whereas σ values are obtained as 0.149 and 0.0015. It is evident that selecting a path with low μ value results in increased value of σ and vice versa. If the entire procedure to determine the desired optimal route is executed using exhaustive search [17], the computational cost is as follows.

$$\begin{aligned} T(C_{r,d}) =& \frac{1}{r} + \frac{4}{r}T(C^{R}_{r-1,*}) + \frac{2(r-2)}{r}T(C^{NR}_{r-1,*}) \\ &+ \frac{2(r-3)}{r}T(C^{NR}_{r-1,*}) \end{aligned} \tag{9}$$

The NIM shows that a radial node $C_{r,d}$ establishes DA with the radial node of its lower level, and contributes only single candidate path, which is the first term in right hand side of (9). If $C_{r,d}$ is non radial, but lies beside any radial node at level r, it shares DA with $(r-2)$ nodes of next lower level. It contributes large number of candidate paths, though such condition occurs for a fixed number of nodes even for high values of r. There is an option to select among radial and non radial nodes in lower level. The former is expressed in the second term and the latter as the third in (9). Here, '$*$' implies that d does not have any influence on this computation. The remaining non radial nodes contributing candidate paths with only IA is shown by the last term of (9). Moreover, traditional exhaustive search takes exponential time to compute $T(C_{r,d})$.

To reduce the computational cost further and scale the proposed solution to larger networks, MOGA is used to search for optimality among the candidates.

Compared to several MOGA based approaches [5], NSGA-II computes the optimal solution in reduced time if the number of objectives is small. Thus, the concept of NSGA-II is introduced in the proposed work to determine the Pareto front with its constraint handling capability. It uses fast non-dominated sorting to evaluate solutions belonging to different fronts. Thus, the execution workflow of finding the optimal route through MOGA based approach is illustrated comprehensively in Figure 5.

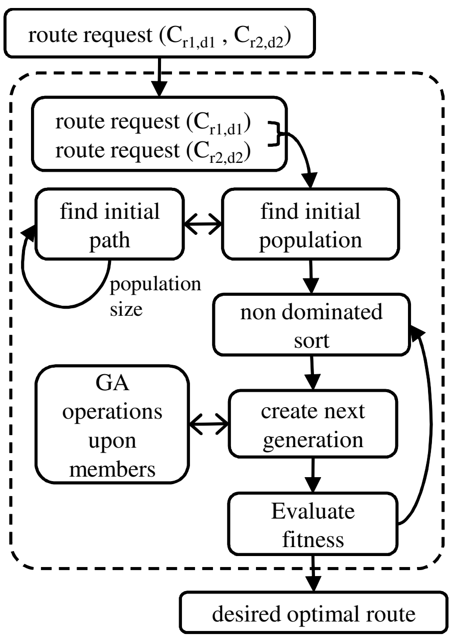

Figure 5: Workflow for finding Optimal Route

As discussed earlier, every request to be scheduled is jointly executed in two parts. For each execution, the initial generation is populated with random routes governed by the population size. Next, this generation is sorted as per non-dominance of objectives. The better half of the generation performs GA operations (selection, crossover and mutation) among themselves to provide the next generation. This cycle is continued till the generation reaches stability in terms of fitness. The desired optimal route is obtained by extracting the solution from the Pareto front corresponding to these two parts followed by subsequent merging.

The proposed MOGA in this work uses a population comprising of a subset of all possible candidate routes between either source node and MSC or MSC and destination node. These routes are considered here as chromosomes, whereas, the nodes and the links in the route denote the genes. Such consideration is necessary to map the network environment onto GA paradigm. The key-points inside the dotted block in Figure 5 are discussed next.

5.2.1 Initial Population

The set of nodes and links in a candidate route is stored as an array to represent a chromosome of the initial population. The size of the initial population follows an adaptive rule and is expressed by (10).

$$\text{Population Size } (P) = \begin{cases} 2^{lim_R} & \text{if } lim_R \leq 6 \\ 100 & \text{otherwise} \end{cases} \quad (10)$$

This is due to the fact that farther a node is from MSC, the number of candidate paths between them is larger. Keeping a fixed population size is useless for smaller networks where the number of candidates is less. The procedure to encode the routes as members of the initial population is illustrated by algorithms 2 and 3.

Algorithm 2 find initial generation ($C_{r,d}$)

paths={};
for population size **do**
  for number of attempts **do**
    path = find initial path(($C_{r,d}$));
    if path $\notin$ paths & constraints followed **then**
      paths = paths $\cup$ {path};
      break loop;
return paths

Algorithm 3 find initial path ($C_{r,d}$)

if r>1 **then**
  cdt={}; // candidates
  $lim_{up} = (r * (r-1) * 3 + 1)$; // upper limit
  $lim_{low} = lim_{up} - (6 * (r-1))$; // lower limit
  for idx1= lim_{low} to lim_{up} **do**
    for idx2: 2 to 7 **do**
      if NIM(idx1,idx2) == $C_{r,d}$ **then**
        cdt=append(cdt,NIM(idx1,idx2));
  subPath = find initial path (random (cdt));
  path = append(subPath, $C_{r,d}$);
else
  path = {MSC,$C_{r,d}$};
return path

5.2.2 Selection, Crossover, and Mutation

After the evaluation of a population, we select an elite set of solutions for the next generation using the fast non-dominated sorting. GA operators are applied upon the elite set to generate offspring which are included in the next generation. Crossover is performed following the spatial property of the candidate paths, i.e. two candidates can perform crossover if they have at least one common node in between. In the proposed work, single-point crossover is performed twice, once between the set of paths connecting source node and MSC, and other between those connecting MSC and destination node. Single point mutation is applied upon the population only where crossover inherently fails to be applied.

In Figure 6 the crossover point between two candidates is highlighted. For the remaining candidates where crossover is not possible, mutation is performed randomly upon a single point in the route. Accordingly node associations are modified. This is illustrated using dotted lines in Figure 6.

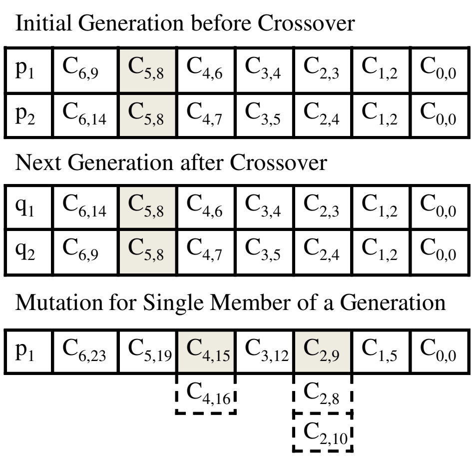

Figure 6: Crossover and mutation in the proposed work

5.2.3 Next Generation from Elite Set

The population of a generation is a combination of both optimal and sub optimal routes between the nodes. These routes are sorted into Pareto fronts using algorithm 4. Each front consists of alternate routes which are non-dominating to each other. The better half of the generation as obtained from the fronts is extracted as the elite set of that generation. GA operations are performed upon this elite set to obtain the offspring set. The elite set along with the offspring set together constitute the next generation as explained by following algorithm 5. In addition, the operator '$\succ$' used in algorithms 4 and 5 denotes the dominance of one solution over the other.

Algorithm 4 non dominated sort(paths)

for path p1 $\in$ paths **do**
    dd_{p1} = {}; //dominated set
    dg_{p1} = 0; // dominating count
    for path p2 $\in$ paths except p1 **do**
        if p2 $\succ$ p1 $\forall$ objective **then**
            dd_{p1} = append(dd_{p1}, p2);
        else if p1 $\succ$ p2 $\forall$ objective **then**
            dg_{p1}= $dg_{p1} + 1$;
    if $dg_{p1} == 0$ **then**
        $F_1 = F_1 \cup \{p1\}$;
current=1;
while $F_{current} \neq \phi$ **do**
    for path $p1 \in F_{current}$ **do**
        for path $p2 \in dd_{p1}$ **do**
            $dg_{p2} = dg_{p2} - 1$;
            if $dg_{p2} == 0$ **then**
                $F_{current+1} = F_{current+1} \cup \{p2\}$;
    current = current +1;

Algorithm 5 create next generation (paths)

es = $\{p \mid p \succ q, q \notin es, |p| = |q| = |paths|/2\}$;
for unselected path p1 $\in$ es **do**
    for unselected path p2 $\in$ es except p1 **do**
        pt = random crossover point (p1, p2);
        if pt $\neq \phi$ **then**
            {q1, q2} = random crossover (p1, p2);
            gen_{next} = gen_{next} $\cup$ {q1, q2};
            $selected_{p1} = selected_{p2} = true$;
for unselected path p1 $\in$ es **do**
    q1 = random single point mutation (p1);
    gen_{next} = gen_{next} $\cup$ {q1};
    $selected_{p1} = true$;
gen_{next} = gen_{next} $\cup$ es;

Observation 1: Crossover and mutation are not simultaneously operated upon a population, as explained in Algorithm 5. Here, mutation is attempted only where crossover is not possible. This is due to the fact that single point mutation, only by itself, fails to introduce significant modification in a candidate route.

5.2.4 Fitness and Stability of Generation

The fitness of each member in a population is obtained using the constraints defined in phase I. Every generation is evaluated for stability. It holds when there is no significant deviation of composite fitness of all members in a generation from that in earlier. Such a generation is termed as a stable one and is subsequently used to obtain the optimal solution from the first Pareto front (F_1), as explained in algorithm 6.

Algorithm 6 route request ($C_{r,d}$)

paths = find initial generation($C_{r,d}$);
cost_array = determine cost $\forall$ path $\in$ paths;
for number of iterations **do**
    paths = non dominated sort(paths);
    paths = create next generation(paths);
    cost_array = determine cost $\forall$ path $\in$ paths;
    if $\triangle$cost_array < predefined limit **then**
        break iteration;
$route_{optimal}$ = {p | p $\in$ paths, p = random(F_1)};
constraint matrices = constraint matrices - (consumption $\in route_{optimal}$);
return $route_{optimal}$;

Observation 2: The computational cost of algorithm 6 is $O(GMP^2)$ where G, M and P denote the number of generations, the number of objectives, and the population size respectively. The P^2 factor should be reduced if possible, since it leads to long processing time for large population size.

5.3 Phase III

Phase II is executed at least once for the set of requests placed in a queue. It may happen that a request from the queue fails to be scheduled and any other scheduled low priority request exists, then the latter can be removed and send back to the request queue. Subsequently, the current request is attempted to be scheduled once again. Thus the concept of priority plays an important role in scheduling the requests as explained by algorithm 7.

Algorithm 7 schedule requests (set of requests in request queue $< C_{r1,d1}$,$C_{r2,d2}$, priority, tolerance$>$)

Initialize NIM and Constraint Matrices;
for request $Req_{current}$$<$$C_{r1,d1}$,$C_{r2,d2}$, Priority, Tolerance$>$ $\in$ request queue (Q) **do**
    path = route request ($C_{r1,d1}$) | route request ($C_{r2,d2}$);
    if path == ϕ **then**
        find request Req_{victim} with $priority_{victim}$ < $priority_{current}$;
        if $Req_{victim} \neq \phi$ **then**
            discard Req_{victim};
            append (Q, Req_{victim});
            path = route request ($C_{r1,d1}$) | route request ($C_{r2,d2}$);
        if path == ϕ **then**
            discard $Req_{current}$;
            append (Q, $Req_{current}$);

Table 2: Used System Parameters

Radius Limit from MSC	20
Allowable unit hop time slot	10ms.
Node Loss constant	0.003
Unit hop loss constant	0.005
User distribution	uniform
User limit per cell	100
Maximum number of iterations in GA	15
Maximum population size in GA	100
Crossover probability	Maximum possible
Mutation probability	Minimum
Simulation total duration	20s.

6 Experimental Results

In this section, the simulation results for various scenarios in order to assess the performance of the proposed model are presented. In this context, the major system parameters are summarized in Table 2. The simulation studies based on these parameter values are carried out in the subsequent sections.

6.1 Variation between constraints

To maximize the number of optimally routed active connection in terms of the call acceptance for the system, the experiment is carried out by modulating either the node capacity or the node association capacity keeping the alternate one at fixed level. In both the scenarios, the system has been stressed under increasing number of scheduling requests, and resulting in acceptable performance. The behaviour of call acceptance against the constraint values is shown in Figure 7. For gradual increment of node association capacity or node capacity, the number of call acceptance is increased for the system. In practice, multiple call requests generated from a node. We have considered the user density level as 2.63, 5.26 and 10.52 by uniformly distributing certain number of users (proportionate to lim_R of the cellular layout considered in simulation) across all the nodes. With the introduction of user density as a parameter, the call acceptance for the system is shown in Figure 8. In experiment, the node association capacity is fixed at 0.1 simulating a stressed system and the corresponding number of call acceptance is shown under such circumstance. Further, the node capacity is also fixed at 0.1

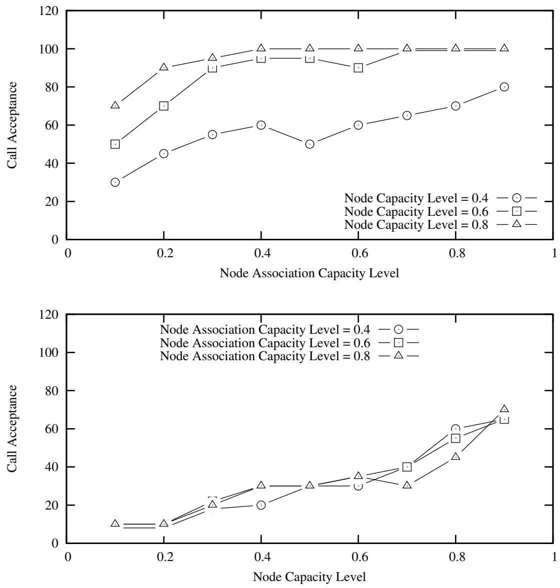

Figure 7: Call Acceptance vs. Constraints (Node Capacity and Node Association Capacity)

for the same reason as in previous, and corresponding variation in call acceptance is plotted. It is observed that improved performance is achieved for gradual increment of either capacity levels.

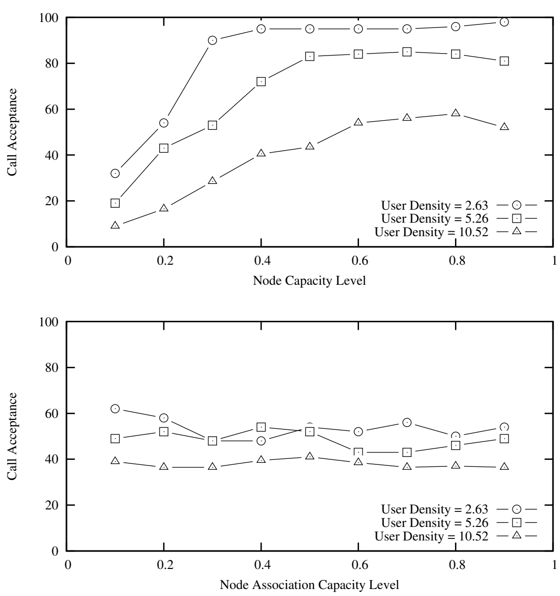

Figure 8: Call Acceptance upon varying User Density

6.2 Interrelation between Objectives

The objectives introduced in the proposed work are plotted for ascertaining the variation of one against the other. It is observed experimentally, that, in accordance with the assumption, these two objectives are conflicting to each other. The MOGA approach towards optimizing both objectives results in a set of Pareto-optimal solutions, as shown in Figure 9, with the help of Pareto fronts. The solutions belonging to separate fronts have a dominating relation between them. However, there is no such overall dominance between the solutions belonging to a single front. After several rounds of iteration of MOGA, we reach to the desired stable fitness of a generation. As discussed in Table 2, the maximum

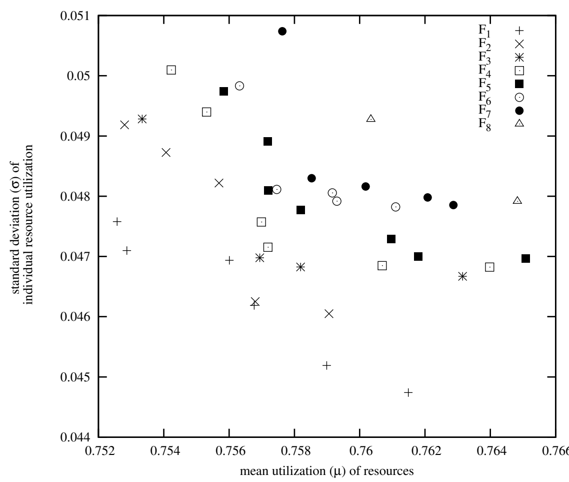

Figure 9: Pareto optimal fronts and solution points

number of iterations of MOGA for obtaining a stable generation in terms of fitness function is fixed to a certain value. This is determined from the observation that the fronts and the solution points belonging to the fronts stabilize after a certain number of steps, and don't vary significantly in further generations. This is experimentally illustrated in Figure 10. We observe that the values of both objectives vary at initial generations; however, they stabilize later on. Thus, after a few generations, an optimal front containing the desired solution is obtained by this approach.

6.3 Performance comparison

Once the stability of the generation becomes certain, the quality of approximation offered in a solution by the members of a generation in MOGA is evaluated. Two standard approaches [5] to ascertain this quality is attempted in this experiment. The former approach (MD)

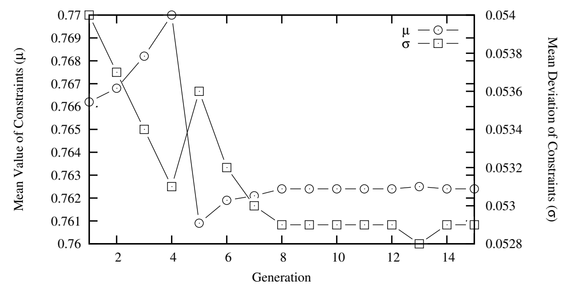

Figure 10: Stabilization of objectives across generations

is involved with consideration of the set of all non-dominated solution points over all the generations as the true Pareto front. Accordingly, comparison of members of each generation in terms of objective functions with the true Pareto front is plotted in Figure 11. Another approach (HI) towards measure of approximation quality is attempted using hypervolume indicator, which is a popular measure of the volume of dominated portion of the objective space related to a reference point. As expected, both the approaches indicate the improvement in quality of approximation with ongoing generations. In addition, the stable convergence of the solutions for the current approach as shown in Figure 11 is obtained with reduced number of generation as compared with the work in [10].

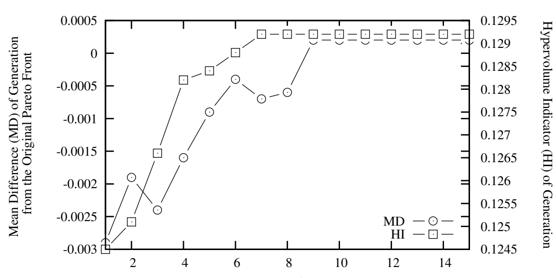

Figure 11: Quality of approximation obtained by proposed MOGA

In order to assess the performance of the proposed model in terms of call acceptance, we compare this work with other end-to-end schedulers [22]. The basic scheme works with a single link model. On the other hand, the joint scheme refers to another end-to-end scheduling method which jointly considers the time varying channel conditions in both directions, i.e., towards MSC and away from MSC. The simulation has been performed by keeping both Node Capacity and Node Association capacity fixed at 0.8 and varying the number of users across nodes of the system. The resulting percentage of calls accepted has been calculated, and is compared against previous two schemes. It is visible from the Figure 12 that the proposed scheme provides improved results in increasing call acceptance.

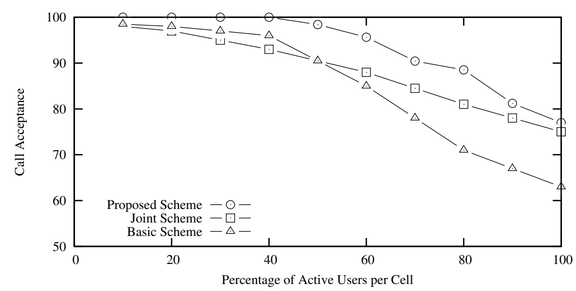

Figure 12: Comparison between proposed and existing schemes

To highlight the importance of the current work, an experiment is carried out where the Node Capacity and Node Asssociation Capacity are simultaneously varied from 10% to 100% of maximum allowable. The result in terms of call acceptance for the proposed model is evaluated and subsequently compared with the declared values of the work in [10]. Hence, the Figure 13 shows an improved performance of the proposed model over such existing approach.

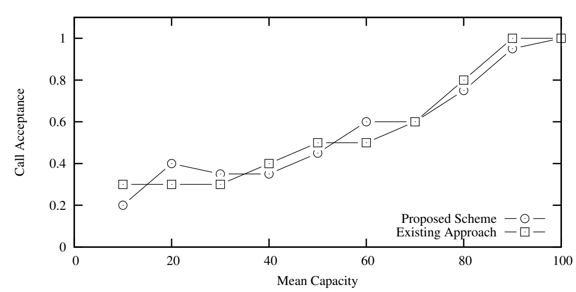

Figure 13: Performance comparison with existing approach

The use of MOGA in the proposed approach has significantly reduced computational complexity when compared with traditional search approaches used in [21]. As shown in Figure 14, the traditional model explodes exponentially when lim_R is increased above a certain value, whereas the proposed approach provides constant growth even for higher values. However, the traditional approach outperforms the proposed approach for initial values of lim_R. The importance of MOGA in the proposed work is highlighted when the solution is attempted for larger networks, where the solution space grows large for high values of r and d according to (9). This approach initiates with few random solutions and subsequently attempts to reach an optimal one without exhaustively searching the entire solution space. Thus the proposed approach exhibits improved computation cost, optimality in routing and higher rate of convergence.

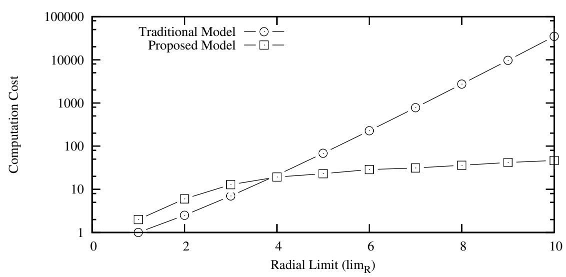

Figure 14: Performance improvement through proposed model

7 Conclusions

The current work proposes a priority based call scheduler featuring joint routing in mobile networks through MOGA based approach. The significance of priority is introduced along with a number of constraints defined over the system. Initially, the cellular layout is represented using NIM followed by the generation of constraint matrices. Then the optimal route for a set of call requests is obtained through MOGA in successive phases. The proposed scheduler is executed and the output is broadcast to all the BSs starting at the MSC. Hence no computation is necessary at the BSs regarding routing.

Performance results demonstrate notable improvements in terms of call acceptance as well as computational cost for the system over other existing approaches. The gain obtained in the proposed model results from optimization in scheduling an individual call while keeping the entire system at reduced congestion as well, which pays off while scheduling the subsequent requests. In addition, the quality of approximation for the proposed model is measured using standard indicators.

The concept of NSGA-II is used in the current work. However, other variations of MOGA can be explored for subsequent improvements. The work can be further enhanced in future by defining the problem with more number of elementary objectives, which can be expected to provide desired solution matching with practical scenarios.

References

[1] Agarwal, R., Majjigi, V. R., Han, Z., Vannithamby, R., and Cioffi, J. M. Low complexity resource allocation with opportunistic feedback over downlink ofdma networks. *IEEE Journal on Selected Areas in Communications*, 26(8):1462–1472, 2008.

[2] Almajano, L. and Perez-Romero, J. Packet scheduling algorithms for interactive and streaming services under qos guarantee in a cdma system. In *Vehicular Technology Conference, 2002. Proceedings. VTC 2002-Fall. 2002 IEEE 56th*, volume 3, pages 1657–1661 vol.3, 2002.

[3] Alpcan, T., Fan, X., Basar, T., Arcak, M., and Wen, J. T. Power control for multicell cdma wireless networks: A team optimization approach. *Wireless Networks*, 14(5):647–657, 2008.

[4] Badia, L., Botta, A., and Lenzini, L. A genetic approach to joint routing and link scheduling for wireless mesh networks. *Ad Hoc Networks*, 7(4):654–664, 2009.

[5] Bringmann, K., Friedrich, T., Neumann, F., and 0007, M. W. Approximation-guided evolutionary multi-objective optimization. In Walsh, T., editor, *IJCAI*, pages 1198–1203. IJCAI/AAAI, 2011.

[6] Camelo, M., Omana, C., and Castro, H. Qos routing algorithm based on multi-objective optimization for wireless mesh networks. In *Communications (LATINCOM), 2010 IEEE Latin-American Conference on*, pages 1–6, Sept 2010.

[7] Changdar, C., Mahapatra, G., and Pal, R. K. An efficient genetic algorithm for multi-objective solid travelling salesman problem under fuzziness. *Swarm and Evolutionary Computation*, 15(0):27 – 37, 2014.

[8] Chiu, H. S. and Yeung, K. L. A joint routing and scheduling algorithm for efficient broadcast in wireless mesh networks. In *WCNC*, pages 1–6. IEEE, 2010.

[9] Chiu, H. S. and Yeung, K. L. Maximizing multicast call acceptance rate in multi-channel multi-interface wireless mesh networks. *Trans. Wireless. Comm.*, 9(8):2622–2631, Aug. 2010.

[10] Cui, X., Lin, C., and Wei, Y. A multiobjective model for qos multicast routing based on genetic algorithm. In *Computer Networks and Mobile*

Computing, 2003. ICCNMC 2003. 2003 International Conference on, pages 49–53, Oct 2003.

[11] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, Apr 2002.

[12] Dianati, M., Shen, X., and Naik, K. Cooperative fair scheduling for the downlink of cdma cellular networks. *IEEE T. Vehicular Technology*, 56(4):1749–1760, 2007.

[13] Fadil, Y. A. Routing using genetic algorithm for large networks. *Diyala Journal of Engineering Sciences*, 3(2):53 – 70, 2010.

[14] Hoffmann, F., Medina, D., and Wolisz, A. Joint routing and scheduling in mobile aeronautical ad hoc networks. *IEEE T. Vehicular Technology*, 62(6):2700–2712, 2013.

[15] Hong-Xia, Y., Liu, X., and Pu-Kun, L. Intelligent algorithms for solving multiobjective optimization problems. In *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on*, pages 1–5, Oct 2008.

[16] Lee, J.-W., Mazumdar, R., and Shroff, N. Joint resource allocation and base-station assignment for the downlink in cdma networks. *Networking, IEEE/ACM Transactions on*, 14(1):1–14, Feb 2006.

[17] Lorenzo, B. and Glisic, S. Optimal routing and traffic scheduling for multihop cellular networks using genetic algorithm. *Mobile Computing, IEEE Transactions on*, 12(11):2274–2288, Nov 2013.

[18] Masazade, E., Rajagopalan, R., Varshney, P., Mohan, C., Sendur, G., and Keskinoz, M. A multiobjective optimization approach to obtain decision thresholds for distributed detection in wireless sensor networks. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 40(2):444–457, April 2010.

[19] Pries, R., Staehle, D., Staehle, B., and Tran-Gia, P. On optimization of wireless mesh networks using genetic algorithms. *International Journal On Advances in Internet Technology*, 1&2, 7 2010.

[20] Raoof, O. and Al-Raweshidy, H. Pricing-based resource allocation in ad-hoc networks using auction theory. In *Future Network and Mobile Summit, 2010*, pages 1–8, June 2010.

[21] Ryu, J., Noel, E. C., and Tang, K. W. Distributed and fault-tolerant routing for borel cayley graphs. *IJDSN*, 2012, 2012.

[22] Saad, W., Dawy, Z., and Sharafeddine, S. A utility-based algorithm for joint uplink/downlink scheduling in wireless cellular networks. *Journal of Network and Computer Applications*, 35(1):348 – 356, 2012. Collaborative Computing and Applications.

[23] Thakurta, P., Poddar, R., and Bandyopadhyay, S. A new approach on co-ordinate based routing protocol for mobile networks. In *Advance Computing Conference (IACC), 2010 IEEE 2nd International*, pages 177–180, Feb 2010.

[24] Wu, Z., Song, H., Jiang, S., and Xu, X. A grid-based stable routing algorithm in mobile ad hoc networks. In *Asia International Conference on Modelling and Simulation*, pages 181–186. IEEE Computer Society, 2007.

[25] Yaipairoj, S. and Harmantzis, F. C. Dynamic pricing with "alternatives" for mobile networks. In *WCNC*, pages 671–676. IEEE, 2004.

[26] Yaipairoj, S. and Harmantzis, F. C. Priority telephony system with pricing alternatives. In de Souza, J. N., Dini, P., and Lorenz, P., editors, *ICT*, volume 3124 of *Lecture Notes in Computer Science*, pages 183–191. Springer, 2004.

[27] Zhengyi, C. and Minyong, L. Price competition dynamic model and its complexity in mobile communication enterprise. In *Management and Service Science, 2009. MASS '09. International Conference on*, pages 1–4, Sept 2009.