Optimization of Scheduling and Routing in Wireless Ad-Hoc Networks Using Cubic Games

Ebrahim Karami[1] and Savo Glisic[2], Senior Member IEEE

[1] Faculty of Engineering and Applied Science, Memorial University of Newfoundland

300 Prince Philip Dr., St. John's, NL A1B 3X5, Canada

[2] Centre for Wireless Communications (CWC), University of Oulu,

P.O. Box 4500, FIN-90014, Oulu, Finland

Abstract— In this paper we present nonlinear 3 player game model for joint routing, network coding, and scheduling problem. To define such a game model, first routing and network coding are modeled by using a new approach based on compressed topology matrix that takes into account the inherent multicast gain of the network. Topology matrix includes the set of all possible paths, including network coded paths, from sources to their corresponding sinks. These paths are identified and compressed, and then by switching between some of them with appropriate usage rates (frequencies), achievable throughput is optimized.

The scheduling is optimized by a new approach called *network graph soft coloring*. Soft graph coloring is designed by switching between different *components* of a wireless network graph, which we refer to as graph fractals, with appropriate usage rates. Therefore each link can be painted with more than one different colors selected with appropriate probabilities. In the proposed game which is a *nonlinear cubic* game, the strategy sets of the players are links, path, and network components. The outputs of this game model are mixed strategy vectors of the second and the third players at equilibrium. Strategy vector of the second player specifies optimum multi-path routing and network coding solution while mixed strategy vector of the third players indicates optimum switching rate among different network components or membership probabilities for optimal soft scheduling approach. Optimum throughput is the value of the proposed *nonlinear cubic* game at equilibrium. The proposed *nonlinear cubic* game is solved by extending fictitious playing method. Numerical and simulation results prove the superior performance of the proposed techniques compared to other conventional schemes.

Keywords— Matrix games, graph fractals, soft graph coloring.

I. INTRODUCTION

Optimization of routing for unicast networks where each source is sending its data to just one sinks, is straightforward and many algorithms are already available on the topic [1, 2]. Multicasting, which is sending data from the source to a group of destinations, provides multicast gain resulting into decreased overall required network resources [3, 4]. In addition to this inherent gain multicasting provides a chance for using network coding gain. Although network coding can be also used in unicasting with multiple sessions, its main advantage is for multicasting. Network coding with additional process at relays and even source nodes increases the overall throughput. This technique, originally proposed for satellite communications, is extended for applications in wired and wireless communications [5]. Although plain routing and network coding usually are not used together, a combination of these techniques can improve the performance [6]. When the problem of joint routing and network coding is extended to wireless ad-hoc network, interference between links must be controlled through scheduling, because of omnidirectional transmissions. Scheduling in wireless communications is modeled as a coloring problem in graph theory where two adjacent areas (sub graphs) cannot be painted with the same color [7]. Therefore the problem of routing and network coding should be solved simultaneously with scheduling [6]. Reference [8] introduces the application of matrix games to find capacity of wired networks with network switching and [6] extends its application to optimize joint routing and network coding for wired networks. In [6] and [9], presentation of graph is modified to path matrix instead of conventional definition by set of nodes and links. Reference [10] uses this problem formulation for joint optimization of routing, and network coding in wireless networks and solves it with a heuristic algorithm.

In this paper a *nonlinear cubic* game is proposed to jointly optimize routing, network coding, multicast gain and scheduling. This new approach optimizes scheduling by introducing concept of soft coloring where any link can be painted with more than one distinct color. The proposed technique although needs to solve a complex non-linear game, presents the optimal solution for the problem. On the other hand, dominancy theory helps us to reduce volume of the cubic payoff matrix.

To solve this game model we propose an extended fictitious play (FP) technique. In FP players can update their belief on other player's strategies based on history of their

978-1-4673-1881-5/12/$31.00 ©2012 IEEE

decisions. This technique which is used to solve matrix games was proposed by Brown [11] and its convergence proved in [12]. FP has originally been proposed for ordinary linear games and in this paper we extend its application for the proposed *non-linear cubic* game model.

The rest of the paper is organized as follows. In Section II, the system model for wireless ad-hoc networks is presented. The matrix game framework combined with soft graph coloring for optimization of link scheduling is presented in Section III. In Section IV, the proposed matrix game model is extended as a *nonlinear cubic* game to jointly optimize routing, network coding, multicast gain and scheduling along with extended FP algorithm to solve the proposed *nonlinear cubic* game is presented. In Section V, numerical results are presented and finally paper is concluded in Section VI.

II. PROBLEM DEFINITION AND SYSTEM MODEL

Assume a multi-source wireless *ad-hoc* network including N nodes. This network is defined as $G(\boldsymbol{V},\boldsymbol{E},\boldsymbol{\zeta},\boldsymbol{s}_1,\boldsymbol{s}_2,\dots,\boldsymbol{s}_M)$ where $\boldsymbol{V}$ is set of nodes with N elements, $\boldsymbol{E}$ is set of L virtual wireless links, $\boldsymbol{\zeta}$ is set of M sources and $\boldsymbol{s}_i$ is set of sinks corresponding to the ith source. If source is sending unicast data, size of its sink set is one. Wireless propagation for this network assumes the following:

1. Omni-directional transmission.
2. Presence of interference due to simultaneous transmission.
3. TDMA as multiple access scheme for different hops without inter time slot interference.

A. Conflict Free Operation

Assume S_{ij} as the power, in dB, required at node j, for transmitting node i to reach the receiving node j at distance d_{ij} with $S_{ij} \propto S_i d_{ij}^{-\alpha}$ where α is attenuation factor and S_i is transmission power of ith node.

By, definition of the conflict free scheduling, any node $k \neq i$, j, receiving the signal from node m, will be interfered by link l_{ij} if and only if $S_{mk} \leq S_{ik} + \beta$, where β, in dB, is acceptable interference margin between two links. In other word, link l_{ij} is adjacent to l_{mk} for any m and any $k \neq i$, j if

$$S_{mk} \leq S_{ik} + \beta. \tag{1}$$

Alternatively, the two links are adjacent if

$$S_{ij} \leqslant S_{mj} + \beta. \tag{2}$$

Whenever l_{ij} and l_{mk} are physically adjacent i.e. they have a common node or either (1) or (2) hold, they cannot be painted by the same color. Using (1) and (2), link adjacency matrix which is used to design network graph coloring algorithm is defined. This type of conflict free transmission needs exact knowledge of the distance between any two nodes in the network.

B. Paths Identification [6, 10]

The next step is to identify the set of all possible paths from each source to its corresponding set of sinks. Unicast path identification can be done using any algorithm available in the literature. For instance Dijkstra algorithm [13] finds shortest path and it can be easily extended to find $I^{m,n}$ shortest path from mth source to its nth corresponding sink.

Assume $P_i^{m,n}$ as ith non-cyclic directed path from the mth source to its nth corresponding sink. Set of all multicast paths from source m to the set of sinks $\mathbf{s}_m$ are calculated as follows,

$$P^m_{i_1 i_2 \cdots i_{|s_m|-1} i_{|s_m|}} = \bigcup_{n=1}^{|s_m|} P^{m,n}_{i_n}, \tag{3}$$

where $\bigcup$ is union and | . | is cardinality operator. We also define network coded paths as sets of links to carry minimum required systematic and parity data from sources to sinks whereas received packets at sinks are decodable.

Next step is compression of the selected paths due to omnidirectional nature of the propagation and inherent multicast gain. If one multicast path consists of two or more links carrying the same information and originating from the same source, they can be substituted with one equivalent link dependent on the definition of the adjacency. If link adjacency is defined based on interference margin, those links are replaced with the one with higher power. If k-hops adjacency definition is used, those links are substituted with one new link. We refer to this process as topology compression that reduces the network interference graph and simplifies the scheduling process.

C. Conventional Scheduling: Motivating Example

In general conflict free scheduling is based on conventional network graph coloring techniques. When conventional graph coloring is used, any link is painted by just one color and no two adjacent links are painted with the same color. When all links have the same capacity and they are used with the same rate, any minimal conventional coloring is optimum solution for links scheduling. Following simple example shows inefficiency of coloring when links must be activated with different usage rates. Assume that we have 3 links l_1 , l_2 , l_3 with usage rates $r_1=3$, $r_2=1$, $r_3=2$. This means that within a given time frame, referred to as clique cycle to be minimized, link l_1 is used during 3 time slots ($r_1=3$), link l_2 during one slot only ($r_2=1$) and link l_3 during 2 slots ($r_3=2$). In addition, due to a given link adjacency, link 1 can be activated with other two simultaneously and two others cannot be activated together. The minimal coloring schemes for these 3 links with their assigned rates are as follows

i) $T_1=\{l_1, l_2\}$ *and* $T_2=\{l_3\}$ required number of time slots (clique cycle) is 5.

ii) $T_1=\{l_1, l_3\}$ *and* $T_2=\{l_2\}$ required number of time slots is 4. Obviously none of these coloring schemes is optimum and the optimum scheduling for these links is,

iii) $T_1=\{l_1, l_2\}$ *during the first time slot and* $T_2=\{l_1, l_3\}$ *during the next two slots;* required number of time slots is 3.

Therefore in optimum solution T_1 and T_2 have non-empty intersection and we call non-overlapped partial topologies as components of the topology represented by *graph fractals*. A topology or network component set is a complete set of links that can be activated at the same time. Any single link is also a component and we call them as first generation network components and τth generation of network components refers to the set of all components with τ members. Network components of each generation are parents of the next generation. For our 3-links example, we have 5 components as follows, $T_1=\{l_1\}$, $T_2=\{l_2\}$, $T_3=\{l_3\}$, $T_4=\{l_1, l_2\}$ *and* $T_5=\{l_1, l_3\}$, where first 3 components are parents of the last two ones. Consequently in general case to optimize the scheduling appropriate network components and their optimal usage rates must be found.

III. MATRIX GAME MODELING FOR OPTIMUM SCHEDULING

A. Formulation of the Game

Given link activation rate $\boldsymbol{r}$ and link capacity $\boldsymbol{c}$ vectors with I elements and network component set ζ with J elements, we define a payoff matrix $\boldsymbol{H}$ as follows,

$$h_{ij}=\begin{cases}\dfrac{c_i}{r_i}, & \text{if } l_i\in\zeta_j\\ 0, & \text{otherwise}\end{cases} \tag{4}$$

where r_i and c_i are activation rate and capacity respectively for ith link and ζ_j is the jth network component. Assume $_{H}$ as payoff matrix of the min-max zero sum game between two players where their strategy sets are links and network components sets respectively. In the sequel we prove that the mixed strategy vector of the second player gives optimum network component rate.

Theorem 1- mixed equilibrium of the zero sum game defined by payoff matrix (4) gives optimum scheduling for usage rate vector $\boldsymbol{r}$.

Proof: Assume $\boldsymbol{x}$ and $\boldsymbol{y}$ as mixed strategies of the players at equilibrium. Since $\boldsymbol{y}$ is normalized to 1, activation of network components in average needs one time slot. Parameter $\tilde{r}_i$ as activation rate of ith link supported by components of vector $\boldsymbol{y}$ is computed as follows,

$$\frac{\tilde{r}_i}{r_i}={}_i[\boldsymbol{H}]\boldsymbol{y} \tag{5}$$

where ${}_i[\boldsymbol{H}]$ is ith row of the $\boldsymbol{H}$. Therefore for a given component rate vector $\boldsymbol{y}$, the link with minimum supported rate, which is bottleneck of the game is,

$$i_{\min imum}=\arg\min{}_i\left({}_i[\boldsymbol{H}]\boldsymbol{y}\right), \tag{6}$$

Consequently optimum component rate vector must maximize $\min_i\left({}_i[\boldsymbol{H}]\boldsymbol{y}\right)$ as

$$\boldsymbol{y}=\arg\max{}_{\mathrm{y}}\min{}_i\left({}_i[\boldsymbol{H}]\boldsymbol{y}\right). \tag{7}$$

And theoretically (7) is equivalent to

$$\boldsymbol{x}=\arg\min{}_x\left(\boldsymbol{x}^T\boldsymbol{H}\boldsymbol{y}\right), \tag{8}$$

$$\boldsymbol{y}=\arg\max{}_{\mathrm{y}}\left(\boldsymbol{x}^T\boldsymbol{H}\boldsymbol{y}\right), \tag{9}$$

where (8) and (9) define equilibrium for game defined by payoff matrix $\boldsymbol{H}$ and value

$$\text{Throughput}=\text{Value}=\max{}_{\mathrm{y}}\min{}_x\left(\boldsymbol{x}^T\boldsymbol{H}\boldsymbol{y}\right). \tag{10}$$

Formulation of the problem as a game gives us the chance to simplify the model using some special properties of the games, like dominancy theory. For instance in general case while solving this game, because of dominancy, components tagged as parent can be ignored because apparently any parent is dominated by its child. Therefore we just need to consider last generation generated (born) from any link. In addition, calculation of mix strategy for linear games is easy and can be performed by fictitious playing method (FP).

IV. MATRIX GAME MODELING FOR JOINT OPTIMUM ROUTING, NETWORK CODING AND SCHEDULING

A. Formulation of the Game

Assume $\boldsymbol{z}$ as vector of path usage rates in the problem of multipath routing. In this case, we modify the proposed game model in Section III by adding a third player with mix strategy vector $\boldsymbol{z}$. Clearly, link usage rates are linear combinations of path usage rates and therefore according to (4), the proposed 3 player game will be defined by modifying parameter r_i in (4) by a linear combination of elements of vector $\boldsymbol{z}$. Consequently, the proposed game in general form becomes *nonlinear cubic* game where strategy sets of its players are links, paths, and network components respectively and the actual payoff and cost received by the first and third players respectively is a nonlinear function of mix strategy chosen by the second player. According to (4), for a given $\boldsymbol{z}$, if the first and the second players choose their ith and jth pure strategy, their payoff and cost will be $1/\sum_{k=1}^{K}a_{ijk}z_k$, where a_{ijk} as elements of 3 dimensional (cubic) matrix is defined as follows,

$$a_{ijq} = \begin{cases} 1, & \text{if } l_i \in P_j, l_i \in \kappa_q \text{ and } P_j \text{ is a compressed multicast path,} \\ \frac{1}{C_c}, & \text{if } l_i \in P_j, l_i \in \kappa_q \text{ and } P_j \text{ is a compressed network coded path,} \\ \infty, & \text{if } l_i \notin P_j, \\ 0, & \text{Otherwise.} \end{cases} \quad (11)$$

and C_c is the number of XOR combined links at the node with network coding. For our general problem of joint optimal modeling of the routing, scheduling, and network coding using game theory, set of payoff matrices introduced in the Section III are replaced by the same number of matrices (this time 3 dimensional) with elements a_{ijk}. The mixed strategies and value of this game are calculated as follows,

$$\boldsymbol{x} = \arg\min_x \sum_{i=1}^{L}\sum_{j=1}^{J} \frac{x_i y_j}{\sum_{k=1}^{K} a_{ijk} z_k}, \quad (12)$$

$$\boldsymbol{y} = \arg\max_y \sum_{i=1}^{L}\sum_{j=1}^{J} \frac{x_i y_j}{\sum_{k=1}^{K} a_{ijk} z_k}, \quad (13)$$

$$\boldsymbol{z} = \arg\max_z \sum_{i=1}^{L}\sum_{j=1}^{J} \frac{x_i y_j}{\sum_{k=1}^{K} a_{ijk} z_k}, \quad (14)$$

$$\text{Throughput=Value=} \min_x \max_y \max_z \sum_{i=1}^{L}\sum_{j=1}^{J} \frac{x_i y_j}{\sum_{j=1}^{J} a_{ijk} z_k}. \quad (15)$$

Since the value of this game is a linear function of mix strategy vector of the first and second player, dominancy theory is still applicable for their strategies but this property of linear games is not valid for the third player's strategies.

Theorem 2- Dominancy theory for the first two players, in the proposed *nonlinear cubic* game: Strategy i_1 of the first player is dominated by its strategy i_2 if for any *j* and *k*, $a_{i_1jk} \leqslant a_{i_2jk}$ and in the same way, strategy j_1 of the second player is dominated by its strategy j_2 if for any *i* and *k*, $a_{ij_1k} \geqslant a_{ij_2k}$.

Proof- We only need to prove the first part of the theorem and the second part is proved in the same way. Assume $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ as strategy vectors at equilibrium. If $a_{i_1jk} \leqslant a_{i_2jk}$ for any *j* and *k*, therefore $\sum_{k=1}^{K} a_{i_1jk} z_k \leqslant \sum_{k=1}^{K} a_{i_2jk} z_k$ and consequently,

$$val = \sum_{i=1}^{L}\sum_{j=1}^{J} \frac{x_i y_j}{\sum_{k=1}^{K} a_{ijk} z_k} = \sum_{\substack{i=1,\, j=1 \\ i\neq i_1, \\ i\neq i_2}}^{L}\sum^{J} \frac{x_i y_j}{\sum_{k=1}^{K} a_{ijk} z_k} + \sum_{j=1}^{J} \frac{x_{i_1} y_j}{\sum_{k=1}^{K} a_{i_1jk} z_k} + \sum_{j=1}^{J} \frac{x_{i_2} y_j}{\sum_{k=1}^{K} a_{i_2jk} z_k}, \quad (16)$$

and,

$$val \geqslant \sum_{\substack{i=1,\, j=1 \\ i\neq i_1, \\ i\neq i_2}}^{L}\sum^{J} \frac{x_i y_j}{\sum_{k=1}^{K} a_{ijk} z_k} + \sum_{j=1}^{J} \frac{(x_{i_1} + x_{i_2}) y_j}{\sum_{k=1}^{K} a_{i_2jk} z_k} \quad (17)$$

Therefore the payoff received by the first player is less than or equal to the value of the game if the player chooses strategy i_2 instead of i_1 and consequently i_1 strategy is dominated by i_2. Inequality is possible iff $x_{i_1} = 0$.

B. Extended Factitious Playing for Nonlinear Cubic Game (20-22)

To solve (12)-(14), we first prove that against any mixed strategies of the second and third players, the best option of the first player is a pure strategy.

Theorem 3- For given mixed strategy vectors $\boldsymbol{y}$ and $\boldsymbol{z}$, first player receives minimum payoff by a pure strategy and for given mixed strategy vectors $\boldsymbol{x}$ and $\boldsymbol{z}$, second player receives maximum payoff by a pure strategy.

Proof- Assume, $i_{\min} = \arg\min_i \sum_{j=1}^{J} \frac{y_j}{\sum_{k=1}^{K} a_{ijk} z_k}$. Consequently we have

$$\sum_{i=1}^{L}\sum_{j=1}^{J} \frac{x_i y_j}{\sum_{k=1}^{K} a_{ijk} z_k} \geqslant \sum_{i=1}^{L}\sum_{j=1}^{J} \frac{x_i y_j}{\sum_{k=1}^{K} a_{i_{min}jk} z_k} = \sum_{i=1}^{L} x_i \sum_{j=1}^{J} \frac{y_j}{\sum_{k=1}^{K} a_{i_{min}jk} z_k} \quad (18)$$

.

Considering $\sum_{i=1}^{I} x_i = 1$, right hand side of (18) is simplified as,

$$\sum_{i=1}^{L}\sum_{j=1}^{J} \frac{x_i y_j}{\sum_{k=1}^{K} a_{ijk} z_k} \geqslant \sum_{j=1}^{J} \frac{y_j}{\sum_{k=1}^{K} a_{i_{min}jk} z_k},. \quad (19)$$

This means payoff received by the first player when it chooses i_{min} th strategy is minimum and proof is complete. The second part of the theorem for the second player is proved in the same way but there is no such a property for the third player and consequently conventional FP algorithm can not be used for the third player.

Algorithm 1: Extending Fictitious Playing Algorithm for nonlinear cubic game (21-23).

Step 1. Initialization of $\boldsymbol{y}$, $\boldsymbol{z}$ as follows,

$q = 0,\ y_j^{(q)} = 0$, for $\forall j \in [2, J]$ and $y_1^{(q)} = 1$, and
$z_k^{(q)} = 0,\ for\ \forall k \in [2, K] and\ z_1^{(q)} = 1.$

Step 2. Setting iteration number $q \leftarrow q+1$ and calculation of the payoff received by different strategies of the first player as,

$$V_{x,i}^{(q)} = \sum_{j=1}^{J} \frac{y_j^{(q-1)}}{\sum_{k=1}^{K} a_{ijk} z_k^{(q-1)}}, \tag{20}$$

Step 3. Choosing best strategy for the first player against $\boldsymbol{y}^{(q)}$ and $\boldsymbol{z}^{(q)}$ as,

$$i_{\min}^{(q)} = \arg\ \min_i V_{x,i}^{(q)}, \tag{21}$$

Step 4. Updating $\boldsymbol{x}^{(q)}$ as

$$x_{i_{\min}^{(q)}}^{(q)} \leftarrow x_{i_{\min}^{(q)}}^{(q-1)} + 1 \ and\ x_i^{(q)} \leftarrow x_i^{(q-1)} \ for\ \forall i \neq i_{\min}^{(q)}).$$

Step 5. Calculation of the payoff received by different strategies of the second player as,

$$V_{y,j}^{(q)} = \sum_{i=1}^{I} \frac{x_i^{(q)}}{\sum_{k=1}^{K} a_{ijk} z_k^{(q-1)}}, \tag{22}$$

Step 6. Choosing best strategy for the second player against $\boldsymbol{x}^{(q)}$ and $\boldsymbol{z}^{(q)}$ as,

$$j_{\max}^{(q)} = \arg\ \max_j V_{y,j}^{(q)}, \tag{23}$$

Step 7. Updating $\boldsymbol{y}^{(q)}$

as $y_{\cdot j_{\max}^{(q)}}^{(q)} \leftarrow y_{j_{\max}^{(q)}}^{(q-1)} + 1 \ and\ y_j^{(q)} \leftarrow y_j^{(q-1)} \ for\ \forall j \neq j_{\max}^{(q)})$.

Step 8. Updating $\boldsymbol{z}^{(q)}$ using following iterative algorithm which is based on steepest descent algorithm. (Notice, value of the game is a non-linear function of z and consequently best strategy of the third player at iteration q can not be found by just maximization over strategy set of this player.)

8.2. Initialization $\boldsymbol{z}_0^{(q)}$ as $\boldsymbol{z}_0^{(q)} = \frac{\boldsymbol{z}^{(q-1)}}{\left|\boldsymbol{z}^{(q-1)}\right|}$ and go to the first iteration t=1.

8.2. Updating $\boldsymbol{z}_t$ as

$$\tilde{z}_{t,k}^{(q)} = z_{t-1,k} - \mu \sum_{i=1}^{I}\sum_{i=1}^{J} \frac{x_i^{(q)} y_j^{(q)} \left(a_{ijk} - a_{ijK}\right)}{\left(\sum_{k=1}^{K} a_{ijk} z_{t-1,k}^{(q)}\right)^2 \sum_{i=1}^{I} x_i^{(q)} \sum_{j=1}^{J} y_j^{(q)}}, \ for\ k \neq K \tag{24}$$

$$\tilde{z}_{t,k}^{(q)} = z_{t-1,k} + \mu \sum_{k=1}^{K-1}\sum_{i=1}^{I}\sum_{i=1}^{J} \frac{x_i^{(q)} y_j^{(q)} \left(a_{ijk} - a_{ijK}\right)}{\left(\sum_{l=1}^{K} a_{ijk} z_{t-1,l}^{(q)}\right)^2 \sum_{i=1}^{I} x_i^{(q)} \sum_{j=1}^{J} y_j^{(q)}}, \ for\ k = K \tag{25}$$

where μ is step size.

8.3. Assuming ρ as minimum of $\tilde{z}_{t,k}$ versus k, $z_{t,k}$ is updated as,

$$z_{t,k} = \begin{cases} \dfrac{\tilde{z}_{t,k}}{\left|\tilde{z}_{t,k}\right|}, & if\ \rho \geq 0 \\[2ex] \dfrac{\tilde{z}_{t,k} - \rho}{\left|\tilde{z}_{t,k} - \rho\right|} & if\ \rho < 0 \end{cases} \tag{26}$$

8.4. $t \leftarrow t+1$, and return to 8.2 for the next iteration. This sub-algorithm is repeated until have convergence.

Step 9. Updating $\boldsymbol{z}^{(q)}$ as $\boldsymbol{z}^{(q)} = \boldsymbol{z}^{(q-1)} + \boldsymbol{z}_t^{(q)}$.

Step 10. Going to step 2, and repeating steps 2-9 to have convergence.

Step 11. After Q iterations when convergence condition holds calculated mixed strategy vector are normalized to one as $\boldsymbol{x}^{(eq)} = \frac{\boldsymbol{x}^{(Q)}}{\left|\boldsymbol{x}^{(Q)}\right|}$, $\boldsymbol{y}^{(eq)} = \frac{\boldsymbol{y}^{(Q)}}{\left|\boldsymbol{y}^{(Q)}\right|}$, and $\boldsymbol{z}^{(eq)} = \frac{\boldsymbol{z}^{(Q)}}{\left|\boldsymbol{z}^{(Q)}\right|}$. This normalization during iterations before convergence is not necessary and can be performed only once after convergence.

V. SIMULATION RESULTS

In this part, *nonlinear cubic* game model presented in Section IV, is simulated over randomly generated graphs with uniformly distributed nodes for the cases N=10 and 20 nodes, and number of sinks between 1 to 5 and results are averaged over 100 independent run. In each run, one node is randomly selected as source and some of other nodes are selected as its corresponding sinks. Then using a modified version of Dijkstra algorithm up to 5 shortest paths from source to each sink are identified and then by combining the identified unicast path using (3) set of multicast paths are constructed. In the next step, wireless links involved in the set of multicast paths are identified and then their adjacency matrix is defined using (1) and (2). Finally the non-*linear cubic game* is defined using (11) and solved by using modified FP algorithm presented in Section IV.B and then the throughput are calculated using (15). Simulation results are compared with single path case with optimum scheduling presented in Section III.

Fig. 1 presents simulation results for N=10 nodes. It can be seen that in all range of interference margin, optimized multipath routing outperformance single path case and this improvement when there are more multicast sinks, is higher. One additional interesting result is throughput floor observed in curves of the proposed multipath routing which proves after settling to these floors, multicast throughput does not decrease anymore with increasing of multicast sinks and this floor can be considered as throughput for broadcasting.

Fig. 2 presents simulation results for N=20 nodes. In this case we can see lower throughput compared to Fig. 6 which is due to higher number of required hops between source and sinks, but still in this case the performance gain compared to single-path routing is a slightly higher than in Fig. 6.

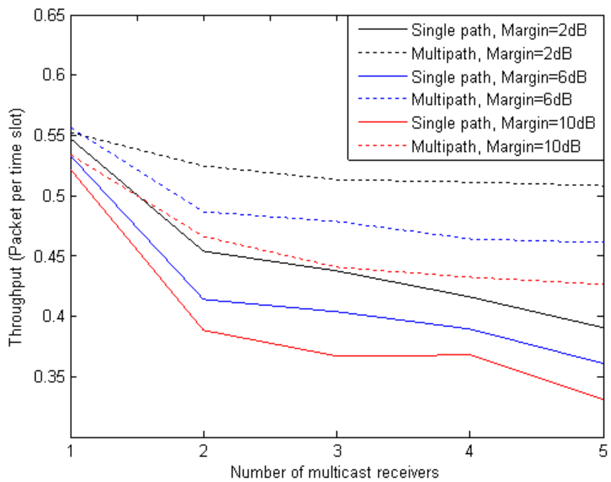

Fig. 1. Multicast throughput of the optimized multipath and single-path routing w.r.t. number of multicast sinks for different values of interference margin when N=10 nodes.

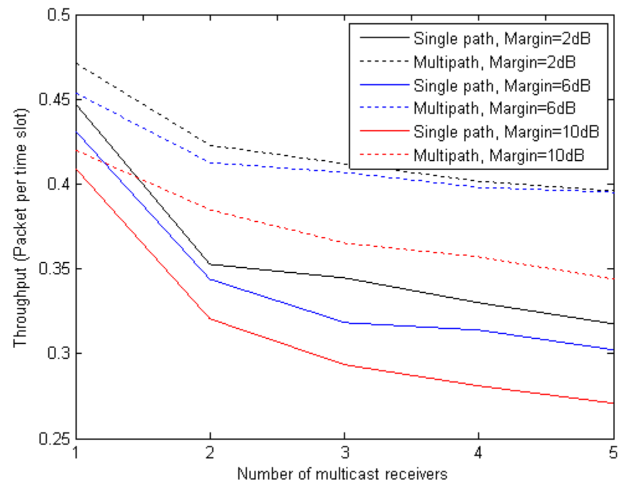

Fig. 2. Multicast throughput of the optimized multipath and single-path routing w.r.t. number of multicast sinks for different values of interference margin when N=20 nodes.

VI. CONCLUSIONS

In this paper novel techniques based on matrix games are proposed to solve joint optimal multipath routing and scheduling problem in wireless *ad-hoc* network. In the first proposed technique *soft scheduling* is modeled by matrix games. To define the game model, wireless topology is first partitioned into networks components, represented by *graph fractals,* which are actually overlapped partial topologies. Then by assigning appropriate usage rates to the selected network components optimum scheduling is achieved. The optimal rates for graph fractals are obtained by calculating mix equilibrium of a matrix game between links and network components. Using matrix games, in this case, helps us to simplify the problem by applying dominancy theory and fictitious playing method, which works iteratively, provides quick convergence.

The proposed optimum scheduling is extended to joint optimum multipath routing and scheduling by adding a third player to the scheduling game. Mixed strategy of this new player at equilibrium optimizes rates assigned to each path. The value of the proposed game is a non-linear function of mixed strategy of the third player and we call this game *nonlinear cubic* game. Because of non-linearity, the proposed game model cannot be solved using conventional FP algorithm and therefore we present an efficient modified FP algorithm to solve this problem. Simulation results prove the efficiency of the proposed game models compared to conventional solutions.

REFERENCES

[1] S. Ramanathan and M. Steenstrup, "A survey of routing techniques for mobile communications networks," Baltzer/ACM *Mobile Networks and Applications*, vol. 1 (1996), pp. 89-104.

[2] X. Hong et al., "Scalable routing protocols for mobile ad hoc networks," *IEEE Network* , Vol. 16, No. 4 (July-Aug. 2002), pp. 11 -21.

[3] J. J. G. L. Aceves and E. L. Madruga, "The Core Assisted Mesh Protocol", *IEEE Journal on Selected Areas in Communications, Special Issue on Ad-Hoc Networks*, vol. 17, no. 8, pages 1380 - 1394, August 1999.

[4] V. Devarapalli, A. A. Selcuk, and D. Sidhu, "MZR: A Multicast Protocol for Mobile Ad Hoc Networks," In *Proc. of the IEEE International Conference on Communications (ICC)*, pages 886 - 891, Helsinki, Finland, June 2001.

[5] R. Ahlswede, N. Cai, S. Li, and R. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[6] E. Karami and S. Glisic, "Joint optimization of network switching and coding using matrix games and network virtualization framework," in proceedings of *the 11th International Symposium on Wireless and Personal Multimedia Communications, WPMC2009,* Sendai, Japan, Sep. 2009.

[7] F. Herrmann and A. Hertz, "Finding the chromatic number by means of critical graphs," *Journal of Experimental Algorithmics*, vol. 7, p. 10, 2002.

[8] Y. E. Sagduyu, and A. Ephremides, "On Joint MAC and Network Coding in Wireless *Ad Hoc* Networks," *IEEE Transactions on Information Theory*, vol. 53, no. 10, pp. 3697-3712, Oct. 2007.

[9] X. B. Liang, "Matrix games in the multicast networks: Maximum information flows with network switching," IEEE Transactions on Information theory, vol. 52, no. 6, pp. 2433-2466, June 2006.

[10] E. Karami and S. Glisic, "Optimization of routing, network coding, and scheduling in wireless multicast Ad-hoc network with topology compression," in proceedings of *the 20th Personal, Indoor and Mobile Radio Communications Symposium, PIMRC2009*.

[11] G.W. Brown, "Iterative solution of games by fictitious play," in *Activity Analysis of Production and Allocation*, T. C. Koopmans, Ed. NewYork: Wiley, 1951, pp. 374–376.

[12] J. Robinson, "An iterative method of solving a game," *Ann. Math.*, vol. 54, no. 2, pp. 296–301, Sep. 1951.

[13] E. W. Dijkstra "A note on two problems in connection with graphs," in *Numerishe Mathematik*, vol. 1, pp. 267-271, 1959.