Maximizing Network Capacity of MPR-Capable Wireless Networks

P.-J. Wan, F. Al-dhelaan, X. Jia, B. Wang, and G. Xing

wan@cs.iit.edu

Outline

- Introduction
- Exam Algorithms in Single Interference Domain
- Approximation Algorithms in General Networks
- PTAS
- Summary

Network Model

- V: networking nodes

Network Model

- V: networking nodes
  - Each $v \in V$ has MPR capability $\tau(v)$

Network Model

- V: networking nodes
  - Each $v \in V$ has MPR capability $\tau(v)$
- A: communication links

Network Model

- V: networking nodes
  - Each $v \in V$ has MPR capability $\tau(v)$
- A: communication links
  - maximum link length $= 1$

Network Model

- V: networking nodes
  - Each $v \in V$ has MPR capability $\tau(v)$
- A: communication links
  - maximum link length $= 1$
  - interference radius $r > 1$

Network Model

- V: networking nodes
  - Each $v \in V$ has MPR capability $\tau(v)$
- A: communication links
  - maximum link length $= 1$
  - interference radius $r > 1$
  - Each $a \in A$ has rate $c(a)$

Independence (i.e. Feasibility) with MPR

Independence (i.e. feasibility) of $I \subseteq A$

1. **MPR Constraint**: Each node v is the receiver of at most $\tau(v)$ links in I.
2. **Interference-free Constraint**: Any two links in I with different receivers are interference-free.

Independence (i.e. Feasibility) with MPR

Independence (i.e. feasibility) of $I \subseteq A$

1. **MPR Constraint**: Each node v is the receiver of at most $\tau(v)$ links in I.
2. **Interference-free Constraint**: Any two links in I with different receivers are interference-free.

$\mathcal{I}$: collection of all independent subsets of A

Max-Weighted Independent Set (MWIS)

Given a non-negative weight function w *on* A*, find an independent subset* I *of* A *with maximum total weight* $\sum_{e \in I} w(e)$.

Max-Weighted Independent Set (MWIS)

Given a non-negative weight function w on A, find an independent subset I of A with maximum total weight $\sum_{e \in I} w(e)$.

Shortest Link Scheduling (SLS)

- A link schedule of $d \in \mathbb{R}_{+}^{A}$:

$$\mathcal{S} = \{(I_j, x_j) \in \mathcal{I} \times \mathbb{R}^{+} : 1 \leq j \leq q\}$$

s.t.

$$d(a) = c(a) \sum_{j=1}^{q} x_j |I_j \cap \{a\}| ;$$

Shortest Link Scheduling (SLS)

- A link schedule of $d \in \mathbb{R}_+^A$:

$$\mathcal{S} = \left\{ \left(I_j, x_j \right) \in \mathcal{I} \times \mathbb{R}^+ : 1 \leq j \leq q \right\}$$

s.t.

$$d\left(a\right) = c\left(a\right) \sum_{j=1}^{q} x_j \left| I_j \cap \left\{ a \right\} \right| ;$$

  - length (or *latency*) of $\mathcal{S}$: $\|\mathcal{S}\| := \sum_{j=1}^{q} x_j$

Shortest Link Scheduling (SLS)

- A link schedule of $d \in \mathbb{R}_+^A$:

$$\mathcal{S} = \{(I_j, x_j) \in \mathcal{I} \times \mathbb{R}^+ : 1 \leq j \leq q\}$$

s.t.

$$d(a) = c(a) \sum_{j=1}^{q} x_j |I_j \cap \{a\}|;$$

  - length (or *latency*) of $\mathcal{S}$: $\|\mathcal{S}\| := \sum_{j=1}^{q} x_j$

- **SLS**: Given a $d \in \mathbb{R}_+^A$, find a shortest link schedule of d.

Shortest Link Scheduling (SLS)

- A link schedule of $d \in \mathbb{R}_{+}^{A}$:

$$\mathcal{S} = \left\{ (I_j, x_j) \in \mathcal{I} \times \mathbb{R}^{+} : 1 \leq j \leq q \right\}$$

s.t.

$$d(a) = c(a) \sum_{j=1}^{q} x_j \left| I_j \cap \{a\} \right|;$$

  - length (or *latency*) of $\mathcal{S}$: $\|\mathcal{S}\| := \sum_{j=1}^{q} x_j$
- **SLS**: Given a $d \in \mathbb{R}_{+}^{A}$, find a shortest link schedule of d.
  - $\chi^{*}(d)$: length of a shortest schedules of d

Shortest Link Scheduling (SLS)

- A link schedule of $d \in \mathbb{R}_+^A$:

$$\mathcal{S} = \left\{ (I_j, x_j) \in \mathcal{I} \times \mathbb{R}^+ : 1 \leq j \leq q \right\}$$

s.t.

$$d(a) = c(a) \sum_{j=1}^{q} x_j \left| I_j \cap \{a\} \right|;$$

  - length (or *latency*) of $\mathcal{S}$: $\|\mathcal{S}\| := \sum_{j=1}^{q} x_j$

- **SLS**: Given a $d \in \mathbb{R}_+^A$, find a shortest link schedule of d.
  - $\chi^*(d)$: length of a shortest schedules of d

- Capacity region of the network:

$$\left\{ d \in \mathbb{R}_+^A : \chi^*(d) \leq 1 \right\}$$

Multiflow

- k unicast requests.

Multiflow

- k unicast requests.
- $\mathcal{F}_j$: the set of flows in (V, E) of the j-th request

Multiflow

- k unicast requests.
- $\mathcal{F}_j$: the set of flows in (V, E) of the j-th request
- A k-flow is a sequence $f = \langle f_1, f_2, \cdots, f_k \rangle$ with $f_j \in \mathcal{F}_j$ $\forall j \in [k]$

Multiflow

- k unicast requests.
- $\mathcal{F}_j$: the set of flows in (V, E) of the j-th request
- A k-flow is a sequence $f = \langle f_1, f_2, \cdots, f_k \rangle$ with $f_j \in \mathcal{F}_j \ \forall j \in [k]$
  - $val(f_j)$: value of f_j

Multiflow

- k unicast requests.
- $\mathcal{F}_j$: the set of flows in (V, E) of the j-th request
- A k-flow is a sequence $f = \langle f_1, f_2, \cdots, f_k \rangle$ with $f_j \in \mathcal{F}_j$ $\forall j \in [k]$
  - $val(f_j)$: value of f_j
  - $\sum_{j=1}^{k} f_j$: cumulative flow of f

Maximum Weighted Multiflow (MWMF)

Given that each request j has a weight $w_j > 0$ per unit of its flow, find a multiflow $f = \langle f_1, \cdots, f_k \rangle$ and a link schedule $\mathcal{S}$ of $\sum_{j=1}^{k} f_j$ such that the length of $\|\mathcal{S}\| \leq 1$ and the total weight of f given by

$$\sum_{j=1}^{k} val\left(f_j\right) w_j.$$

is maximized.

Maximum Concurrent Multiflow (MCMF)

Given that each request j has a demand $d_j > 0$, find a multiflow $f = \langle f_1, \cdots, f_k \rangle$ and a link schedule $\mathcal{S}$ of $\sum_{j=1}^{k} f_j$ such that $\|\mathcal{S}\| \leq 1$ and the concurrency of f given by

$$\min_{1 \leq j \leq k} val\left(f_j\right) / d_j.$$

is maximized.

Algorithmic Issues

- At least as hard as those without MPR capability (i.e., $\tau(v) = 1$ $\forall v \in V$)

Algorithmic Issues

- At least as hard as those without MPR capability (i.e., $\tau(v) = 1$ $\forall v \in V$)
- Non-applicability of traditional graph-theoretic techniques due to the non-binary nature of the link independence

Algorithmic Issues

- At least as hard as those without MPR capability (i.e., $\tau(v) = 1$ $\forall v \in V$)
- Non-applicability of traditional graph-theoretic techniques due to the non-binary nature of the link independence
- Still the same approximality?

Roadmap

- Introduction
- **Exam Algorithms in Single Interference Domain**
- Approximation Algorithms in General Networks
- PTAS
- Summary

Exact Algorithm for MWIS

- A set of $\min\left\{\left|\delta^{in}(v)\right|, \tau(v)\right\}$ heaviest links in $\delta^{in}(v)$ for some $v \in V$.

Exact Algorithm for MWIS

- A set of $\min\left\{\left|\delta^{in}(v)\right|, \tau(v)\right\}$ heaviest links in $\delta^{in}(v)$ for some $v \in V$.
- Enumeration

Exact Algorithm for SLS

- B: links with positive demands

Exact Algorithm for SLS

- B: links with positive demands
- Concatenation of SLSs of the demands by $\delta_B^{in}(v)$ for all $v \in V$

Exact Algorithm for SLS

- B: links with positive demands
- Concatenation of SLSs of the demands by $\delta_B^{in}(v)$ for all $v \in V$
- SLS of the demands by $\delta_B^{in}(v)$

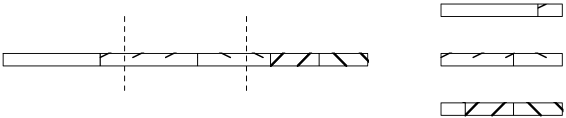
Exact Algorithm for SLS

- B: links with positive demands
- Concatenation of SLSs of the demands by $\delta_B^{in}(v)$ for all $v \in V$
- SLS of the demands by $\delta_B^{in}(v)$
  - Length

$$\max\left\{\max_{a\in\delta_B^{in}(v)} \frac{d(a)}{c(a)}, \frac{\sum_{a\in\delta_B^{in}(v)} \frac{d(a)}{c(a)}}{\tau(v)}\right\}.$$

Exact Algorithm for SLS

- B: links with positive demands
- Concatenation of SLSs of the demands by $\delta_B^{in}(v)$ for all $v \in V$
- SLS of the demands by $\delta_B^{in}(v)$
  - Length
$$\max\left\{\max_{a\in\delta_B^{in}(v)}\frac{d(a)}{c(a)},\frac{\sum_{a\in\delta_B^{in}(v)}\frac{d(a)}{c(a)}}{\tau(v)}\right\}.$$
  - A wrap-around scheme


(a) (b)

Capacity Region

$$\Omega = \left\{ d \in \mathbb{R}_+^A : \sum_{v \in V} \max \left\{ \max_{a \in \delta^{in}(v)} \frac{d(a)}{c(a)}, \frac{\sum_{a \in \delta^{in}(v)} \frac{d(a)}{c(a)}}{\tau(v)} \right\} \leq 1 \right\}.$$

Exact Algorithms for MWMF, MCMF

MWMF:

$$
\begin{array}{ll}
\max & \sum_{j=1}^{k} w_j \cdot val\left(f_j\right) \\
s.t. & f_j \in \mathcal{F}_j, \forall 1 \leq j \leq k; \\
 & \sum_{j=1}^{k} f_j \in \Omega.
\end{array}
$$

Exact Algorithms for MWMF, MCMF

MWMF:

$$\begin{aligned} \max \quad & \sum_{j=1}^{k} w_j \cdot val\left(f_j\right) \\ s.t. \quad & f_j \in \mathcal{F}_j, \forall 1 \leq j \leq k; \\ & \sum_{j=1}^{k} f_j \in \Omega. \end{aligned}$$

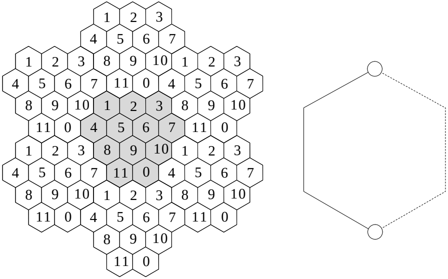
MCMF:

$$\begin{aligned} \max \quad & \phi \\ s.t. \quad & f_j \in \mathcal{F}_j, \forall 1 \leq j \leq k; \\ & val\left(f_j\right) \geq \phi d_j, \forall 1 \leq i \leq k; \\ & \sum_{j=1}^{k} f_j \in \Omega. \end{aligned}$$

Roadmap

- Introduction
- Exam Algorithms in Single Interference Domain
- **Approximation Algorithms in General Networks**
- PTAS
- Summary

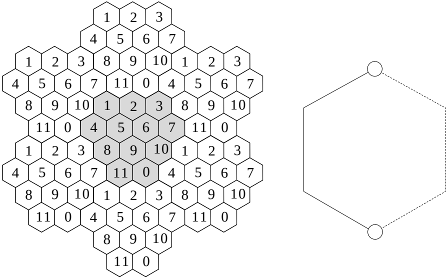
Spatial Division



(a) (b)

- hexagon cells of diameter $r - 1$

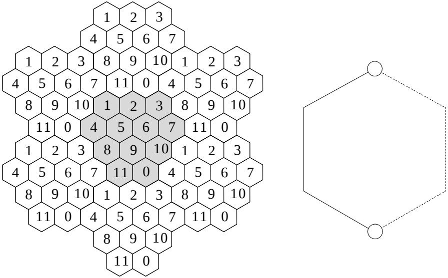
Spatial Division



(a) (b)

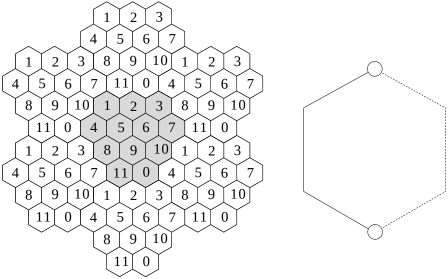
- hexagon cells of diameter $r - 1$
- associate a link with a cell by receiver

Spatial Division



(a) (b)

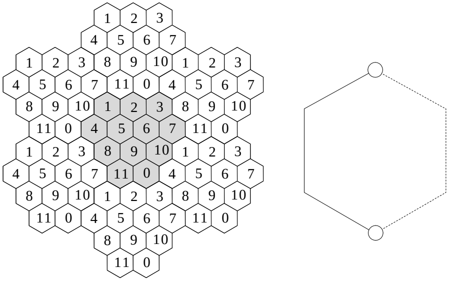
- hexagon cells of diameter $r - 1$
- associate a link with a cell by receiver
- cell labelling: all co-label cells are apart $> r + 1$



(a) (b)

- hexagon cells of diameter $r - 1$
- associate a link with a cell by receiver
- cell labelling: all co-label cells are apart $> r + 1$
    - $\#$ of labels $7 \leq \lambda \leq 12$ for $r \geq 3$

Spatial Division



(a) (b)

- hexagon cells of diameter $r - 1$
- associate a link with a cell by receiver
- cell labelling: all co-label cells are apart $> r + 1$
  - # of labels $7 \leq \lambda \leq 12$ for $r \geq 3$
- $\forall 0 \leq i < \lambda$, $\mathcal{S}_i \leftarrow$ the set of non-empty cells with label i

Divide-And-Conquer Algorithm for MWIS

- **Conquer**: For each non-empty cell S, $I_S \leftarrow$ a MWIS of all links associated with S.

Divide-And-Conquer Algorithm for MWIS

- **Conquer**: For each non-empty cell S, $I_S \leftarrow$ a MWIS of all links associated with S.
- **Combination**: Output the heaviest one among

$$\left\{\bigcup_{S\in\mathcal{S}_i} I_S : 0 \leq i < \lambda\right\}$$

Divide-And-Conquer Algorithm for MWIS

- **Conquer**: For each non-empty cell S, $I_S \leftarrow$ a MWIS of all links associated with S.
- **Combination**: Output the heaviest one among

$$\left\{\bigcup_{S \in \mathcal{S}_i} I_S : 0 \leq i < \lambda\right\}$$

- λ-approixmate

Divide-And-Conquer Algorithm for SLS

- **Conquer**: $\forall$non-empty cell S, $\Gamma_S \leftarrow$ a SLS of the demands by links associated with S.

Divide-And-Conquer Algorithm for SLS

- **Conquer**: $\forall$non-empty cell S, $\Gamma_S \leftarrow$ a SLS of the demands by links associated with S.
- **Combination**:

Divide-And-Conquer Algorithm for SLS

- **Conquer**: $\forall$non-empty cell S, $\Gamma_S \leftarrow$ a SLS of the demands by links associated with S.
- **Combination**:
  - **Merge**: $\forall 0 \leq i < \lambda$, $\Pi_i \leftarrow$ merged schedule from Γ_S for all $S \in \mathcal{S}_i$

Divide-And-Conquer Algorithm for SLS

- **Conquer**: $\forall$non-empty cell S, $\Gamma_S \leftarrow$ a SLS of the demands by links associated with S.
- **Combination**:
  - **Merge**: $\forall 0 \leq i < \lambda$, $\Pi_i \leftarrow$ merged schedule from Γ_S for all $S \in \mathcal{S}_i$
  - **Concatenation**: Output $\Pi \leftarrow$concatenation of Π_i for $0 \leq i < \lambda$

$$\|\Pi\| = \sum_{i=0}^{\lambda-1} \max_{S \in \mathcal{S}_i} \sum_{v \in V_S} \max \left\{ \max_{a \in \delta^{in}(v)} \frac{d(a)}{c(a)}, \frac{\sum_{a \in \delta^{in}(v)} \frac{d(a)}{c(a)}}{\tau(v)} \right\}$$

Divide-And-Conquer Algorithm for SLS

- **Conquer**: $\forall$non-empty cell S, $\Gamma_S \leftarrow$ a SLS of the demands by links associated with S.
- **Combination**:
  - **Merge**: $\forall 0 \leq i < \lambda$, $\Pi_i \leftarrow$ merged schedule from Γ_S for all $S \in \mathcal{S}_i$
  - **Concatenation**: Output $\Pi \leftarrow$concatenation of Π_i for $0 \leq i < \lambda$

$$\|\Pi\| = \sum_{i=0}^{\lambda-1} \max_{S \in \mathcal{S}_i} \sum_{v \in V_S} \max \left\{ \max_{a \in \delta^{in}(v)} \frac{d(a)}{c(a)}, \frac{\sum_{a \in \delta^{in}(v)} \frac{d(a)}{c(a)}}{\tau(v)} \right\}$$

- λ-approximate solution.

Approximate Capacity Region

$$\Phi = \left\{ d \in \mathbb{R}_+^A : \sum_{i=0}^{\lambda-1} \max_{S \in \mathcal{S}_i} \sum_{v \in V_S} \max \left\{ \max_{a \in \delta^{in}(v)} \frac{d(a)}{c(a)}, \frac{\sum_{a \in \delta^{in}(v)} \frac{d(a)}{c(a)}}{\tau(v)} \right\} \leq 1 \right\}.$$

λ-approximate:

$$\Phi \subseteq \Omega \subseteq \lambda \Phi.$$

Exact Restricted Mutiflows

Φ-restricted MWMF:

$$\begin{array}{ll} \max & \sum_{j=1}^{k} w_j \cdot val\left(f_j\right) \\ s.t. & f_j \in \mathcal{F}_j, \forall 1 \leq j \leq k; \\ & \sum_{j=1}^{k} f_j \in \Phi. \end{array}$$

Φ-restricted MWMF:

$$\begin{array}{ll} \max & \sum_{j=1}^{k} w_j \cdot val\left(f_j\right) \\ s.t. & f_j \in \mathcal{F}_j, \forall 1 \leq j \leq k; \\ & \sum_{j=1}^{k} f_j \in \Phi. \end{array}$$

Φ-restricted MCMF:

$$\begin{array}{ll} \max & \phi \\ s.t. & f_j \in \mathcal{F}_j, \forall 1 \leq j \leq k; \\ & val\left(f_j\right) \geq \phi d_j, \forall 1 \leq i \leq k; \\ & \sum_{j=1}^{k} f_j \in \Phi. \end{array}$$

Exact Restricted Mutiflows

Φ-restricted MWMF:

$$\begin{array}{ll} \max & \sum_{j=1}^{k} w_j \cdot val\left(f_j\right) \\ s.t. & f_j \in \mathcal{F}_j, \forall 1 \leq j \leq k; \\ & \sum_{j=1}^{k} f_j \in \Phi. \end{array}$$

Φ-restricted MCMF:

$$\begin{array}{ll} \max & \phi \\ s.t. & f_j \in \mathcal{F}_j, \forall 1 \leq j \leq k; \\ & val\left(f_j\right) \geq \phi d_j, \forall 1 \leq i \leq k; \\ & \sum_{j=1}^{k} f_j \in \Phi. \end{array}$$

λ-approximations

Roadmap

- Introduction
- Exam Algorithms in Single Interference Domain
- Approximation Algorithms in General Networks
- PTAS
- Summary

PTAS

- Constant-bounded the maximum MPR capability $\overline{\tau} := \max_{v \in V} \tau(v)$

PTAS

- Constant-bounded the maximum MPR capability $\overline{\tau} := \max_{v \in V} \tau(v)$
- # of independent links whose interference ranges contained in square of side L is at most Then,

$$|I| \leq \frac{4\left(\left\lceil \pi / \arcsin \frac{r-1}{2r} \right\rceil - 1\right)\overline{\tau}}{\pi} L^2.$$

PTAS

- Constant-bounded the maximum MPR capability $\overline{\tau} := \max_{v \in V} \tau(v)$
- # of independent links whose interference ranges contained in square of side L is at most Then,

$$|I| \leq \frac{4\left(\left\lceil \pi / \arcsin \frac{r-1}{2r} \right\rceil - 1\right)\overline{\tau}}{\pi} L^2.$$

- PTAS for **MWIS**: shifting + dynamic programming

PTAS

- Constant-bounded the maximum MPR capability $\overline{\tau} := \max_{v \in V} \tau(v)$
- $\#$ of independent links whose interference ranges contained in square of side L is at most Then,

$$|I| \leq \frac{4 \left(\left\lceil \pi / \arcsin \frac{r-1}{2r} \right\rceil - 1 \right) \overline{\tau}}{\pi} L^2.$$

- PTAS for **MWIS**: shifting + dynamic programming
- PTAS for **SLS**, **MWMF**, and **MCMF**: approximation-preserving reductions from **MWIS**

Roadmap

- Introduction
- Exam Algorithms in Single Interference Domain
- Approximation Algorithms in General Networks
- PTAS
- **Summary**

Summary

- Exam Algorithms in Single Interference Domain
- Approximation Algorithms in General Networks
- PTAS
- Future work: arbitrary interference radii