Partial Degree Bounded Edge Packing Problem with Arbitrary Bounds

Pawan Aurora   Sumit Singh   Shashank K. Mehta

Indian Institute of Technology, Kanpur

FAW-AAIM 2013

Problem Definition

PDBEP problem

- Input: Graph $G = (V, E)$ and degree-bound function $c : V \to \mathbb{Z}^*$.

Problem Definition

PDBEP problem

- Input: Graph $G = (V, E)$ and degree-bound function $c : V \to \mathbb{Z}^*$.
- Objective: Compute a maximum cardinality set $E' \subseteq E$ which satisfies the *degree condition*:
  $(d'_u \leq c_u) \vee (d'_v \leq c_v)$ for each $e = (u, v) \in E'$.

Problem Definition

PDBEP problem

- Input: Graph $G = (V, E)$ and degree-bound function $c : V \to \mathbb{Z}^*$.
- Objective: Compute a maximum cardinality set $E' \subseteq E$ which satisfies the *degree condition*: $(d'_u \le c_u) \vee (d'_v \le c_v)$ for each $e = (u, v) \in E'$.
- Introduced by Peng Zhang in FAW-AAIM 2012.

Problem Definition

PDBEP problem

- Input: Graph $G = (V, E)$ and degree-bound function $c : V \to \mathbb{Z}^*$.
- Objective: Compute a maximum cardinality set $E' \subseteq E$ which satisfies the *degree condition*: $(d'_u \leq c_u) \vee (d'_v \leq c_v)$ for each $e = (u, v) \in E'$.
- Introduced by Peng Zhang in FAW-AAIM 2012.
- Zhang showed PDBEP is NP-Hard even for uniform $c_v = 1$.

Our Contribution

Zhang, 2012

For uniform $c_v = k$:

- 2-approx for $k = 1$.

Our Contribution

Zhang, 2012

For uniform $c_v = k$:

- 2-approx for $k = 1$.
- $\frac{32}{11}$-approx for $k = 2$.

Our Contribution

Zhang, 2012

For uniform $c_v = k$:

- 2-approx for $k = 1$.
- $\frac{32}{11}$-approx for $k = 2$.
- $O(n^2)$-time exact algorithm on trees for any k.

Our Contribution

Zhang, 2012

For uniform $c_v = k$:

- 2-approx for $k = 1$.
- $\frac{32}{11}$-approx for $k = 2$.
- $O(n^2)$-time exact algorithm on trees for any k.

Our Results

For arbitrary $c : V \to \mathbb{Z}^*$:

Our Contribution

Zhang, 2012

For uniform $c_v = k$:

- 2-approx for $k = 1$.
- $\frac{32}{11}$-approx for $k = 2$.
- $O(n^2)$-time exact algorithm on trees for any k.

Our Results

For arbitrary $c : V \to \mathbb{Z}^*$:

- 2-approx.

Our Contribution

Zhang, 2012

For uniform $c_v = k$:

- 2-approx for $k = 1$.
- $\frac{32}{11}$-approx for $k = 2$.
- $O(n^2)$-time exact algorithm on trees for any k.

Our Results

For arbitrary $c : V \to \mathbb{Z}^*$:

- 2-approx.
- $1.5/(1-\epsilon)$-approx for large values of c_v/d_v.

Our Contribution

Zhang, 2012

For uniform $c_v = k$:

- 2-approx for $k = 1$.
- $\frac{32}{11}$-approx for $k = 2$.
- $O(n^2)$-time exact algorithm on trees for any k.

Our Results

For arbitrary $c : V \to \mathbb{Z}^*$:

- 2-approx.
- $1.5/(1-\epsilon)$-approx for large values of c_v/d_v.
- $O(n \log n)$-time exact algorithm on trees.

Our Contribution

Zhang, 2012

For uniform $c_v = k$:

- 2-approx for $k = 1$.
- $\frac{32}{11}$-approx for $k = 2$.
- $O(n^2)$-time exact algorithm on trees for any k.

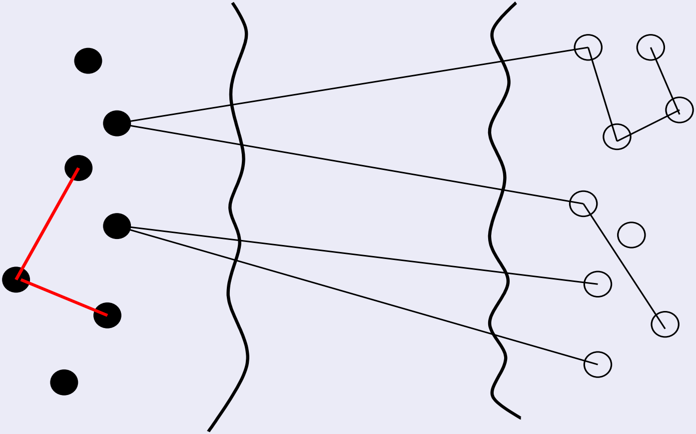
Our Results

For arbitrary $c : V \to \mathbb{Z}^*$:

- 2-approx.
- $1.5/(1-\epsilon)$-approx for large values of c_v/d_v.
- $O(n \log n)$-time exact algorithm on trees.
- $2 + \log_2 n$-approx with weights on edges.

Upper Bound

Upper Bound on OPT



$OPT \leq \sum_{w \in W} c_w \leq \sum_{v \in V} c_v$, where $W = \{v \in V | d'_v \leq c_v\}$.

2-Approximation

Edge Deletion based Algorithm

- Iteratively delete violating edges.

2-Approximation

Edge Deletion based Algorithm

- Iteratively delete violating edges.
- Output the set Y of surviving edges.

2-Approximation

Edge Deletion based Algorithm

- Iteratively delete violating edges.
- Output the set Y of surviving edges.
- Observe that $d_Y(v) \geq c_v \; \forall \; v \in V$.

2-Approximation

Edge Deletion based Algorithm

- Iteratively delete violating edges.
- Output the set Y of surviving edges.
- Observe that $d_Y(v) \geq c_v \; \forall \; v \in V$.
- Hence $|Y| \geq \sum_v c_v/2 \geq OPT/2$.

Weighted PDBEP

$2 + \log_2 n$-Approximation

- Let $H(v)$ be the heaviest c_v edges incident on vertex v.

Weighted PDBEP

$2 + \log_2 n$-Approximation

- Let $H(v)$ be the heaviest c_v edges incident on vertex v.
- Clearly $OPT \leq \sum_{v \in V} \sum_{e \in H(v)} w(e)$.

Weighted PDBEP

$2 + \log_2 n$-Approximation

- Let $H(v)$ be the heaviest c_v edges incident on vertex v.
- Clearly $OPT \leq \sum_{v \in V} \sum_{e \in H(v)} w(e)$.
- Idea is to construct upto $1 + \log_2 |V|$ solutions, which cover $\cup_{v \in V} H(v)$.

Weighted PDBEP

$2 + \log_2 n$-Approximation

- Let $H(v)$ be the heaviest c_v edges incident on vertex v.
- Clearly $OPT \leq \sum_{v \in V} \sum_{e \in H(v)} w(e)$.
- Idea is to construct upto $1 + \log_2 |V|$ solutions, which cover $\cup_{v \in V} H(v)$.
- Then the heaviest solution gives a $2 + \log_2 |V|$ approximation of the problem.

Input Graph is an Unweighted Tree

Exact Algorithm

- At each vertex v we maintain values of three solutions of $T(v)$.

Input Graph is an Unweighted Tree

Exact Algorithm

- At each vertex v we maintain values of three solutions of $T(v)$.
- The algorithm initializes these values to zero for the leaf vertices and computes these values for the internal vertices bottom up.

Input Graph is an Unweighted Tree

Exact Algorithm

- At each vertex v we maintain values of three solutions of $T(v)$.
- The algorithm initializes these values to zero for the leaf vertices and computes these values for the internal vertices bottom up.
- Finally it outputs the maximum of the three values of the root R.

Input Graph is an Unweighted Tree

Exact Algorithm

- At each vertex v we maintain values of three solutions of $T(v)$.
- The algorithm initializes these values to zero for the leaf vertices and computes these values for the internal vertices bottom up.
- Finally it outputs the maximum of the three values of the root R.
- In order to compute the values for internal nodes we need to sort the child nodes with respect to some key values.

Input Graph is an Unweighted Tree

Exact Algorithm

- At each vertex v we maintain values of three solutions of $T(v)$.
- The algorithm initializes these values to zero for the leaf vertices and computes these values for the internal vertices bottom up.
- Finally it outputs the maximum of the three values of the root R.
- In order to compute the values for internal nodes we need to sort the child nodes with respect to some key values.
- Thus at each vertex we incur $O(|Ch| \log |Ch|)$ cost.

Input Graph is an Unweighted Tree

Exact Algorithm

- At each vertex v we maintain values of three solutions of $T(v)$.
- The algorithm initializes these values to zero for the leaf vertices and computes these values for the internal vertices bottom up.
- Finally it outputs the maximum of the three values of the root R.
- In order to compute the values for internal nodes we need to sort the child nodes with respect to some key values.
- Thus at each vertex we incur $O(|Ch| \log |Ch|)$ cost.
- That gives an overall time complexity of $O(n \log n)$.

Natural IP

The natural IP formulation of the problem

IP1: max $\psi = \sum_{e \in E} y_e$, subject to

Natural IP

The natural IP formulation of the problem

IP1: max $\psi = \sum_{e \in E} y_e$, subject to

1. $y_e \leq x_u + x_v \; \forall e = (u, v) \in E$.

Natural IP

The natural IP formulation of the problem

IP1: max $\psi = \sum_{e \in E} y_e$, subject to

1. $y_e \leq x_u + x_v \ \forall e = (u, v) \in E$.
2. $\sum_{e \in \delta(v)} y_e \leq c_v x_v + d_v(1 - x_v) \ \forall v \in V$.

Natural IP

The natural IP formulation of the problem

IP1: max $\psi = \sum_{e \in E} y_e$, subject to

1. $y_e \leq x_u + x_v \ \forall e = (u, v) \in E$.
2. $\sum_{e \in \delta(v)} y_e \leq c_v x_v + d_v(1 - x_v) \ \forall v \in V$.
3. $x_v \in \{0, 1\} \ \forall v \in V$, $y_e \in \{0, 1\} \ \forall e \in E$.

Natural IP

The natural IP formulation of the problem

IP1: max $\psi = \sum_{e \in E} y_e$, subject to

1. $y_e \leq x_u + x_v \; \forall e = (u, v) \in E$.
2. $\sum_{e \in \delta(v)} y_e \leq c_v x_v + d_v(1 - x_v) \; \forall v \in V$.
3. $x_v \in \{0, 1\} \; \forall v \in V$, $y_e \in \{0, 1\} \; \forall e \in E$.

IP1 has large Integrality Gap

Natural IP

The natural IP formulation of the problem

IP1: max $\psi = \sum_{e \in E} y_e$, subject to

1. $y_e \leq x_u + x_v \; \forall e = (u, v) \in E$.
2. $\sum_{e \in \delta(v)} y_e \leq c_v x_v + d_v(1 - x_v) \; \forall v \in V$.
3. $x_v \in \{0, 1\} \; \forall v \in V$, $y_e \in \{0, 1\} \; \forall e \in E$.

IP1 has large Integrality Gap

- Consider K_n with uniform degree constraint $c_v = 1$.

Natural IP

The natural IP formulation of the problem

IP1: max $\psi = \sum_{e \in E} y_e$, subject to

1. $y_e \leq x_u + x_v \; \forall e = (u, v) \in E$.
2. $\sum_{e \in \delta(v)} y_e \leq c_v x_v + d_v(1 - x_v) \; \forall v \in V$.
3. $x_v \in \{0, 1\} \; \forall v \in V$, $y_e \in \{0, 1\} \; \forall e \in E$.

IP1 has large Integrality Gap

- Consider K_n with uniform degree constraint $c_v = 1$.
- Set $x_v = 0.5 \; \forall \; v$. Consider disjoint cliques on $\lfloor n/2 \rfloor + 1$ and $n - \lfloor n/2 \rfloor - 1$ vertices. Set $y_e = 1$ for edges in these cliques.

Natural IP

The natural IP formulation of the problem

IP1: max $\psi = \sum_{e \in E} y_e$, subject to

1. $y_e \leq x_u + x_v \; \forall e = (u, v) \in E$.
2. $\sum_{e \in \delta(v)} y_e \leq c_v x_v + d_v(1 - x_v) \; \forall v \in V$.
3. $x_v \in \{0, 1\} \; \forall v \in V$, $y_e \in \{0, 1\} \; \forall e \in E$.

IP1 has large Integrality Gap

- Consider K_n with uniform degree constraint $c_v = 1$.
- Set $x_v = 0.5 \; \forall \; v$. Consider disjoint cliques on $\lfloor n/2 \rfloor + 1$ and $n - \lfloor n/2 \rfloor - 1$ vertices. Set $y_e = 1$ for edges in these cliques.
- $\psi \geq (n-1)^2/4$.

Natural IP

The natural IP formulation of the problem

IP1: max $\psi = \sum_{e \in E} y_e$, subject to

1. $y_e \leq x_u + x_v \ \forall e = (u, v) \in E$.
2. $\sum_{e \in \delta(v)} y_e \leq c_v x_v + d_v(1 - x_v) \ \forall v \in V$.
3. $x_v \in \{0, 1\} \ \forall v \in V$, $y_e \in \{0, 1\} \ \forall e \in E$.

IP1 has large Integrality Gap

- Consider K_n with uniform degree constraint $c_v = 1$.
- Set $x_v = 0.5 \ \forall \ v$. Consider disjoint cliques on $\lfloor n/2 \rfloor + 1$ and $n - \lfloor n/2 \rfloor - 1$ vertices. Set $y_e = 1$ for edges in these cliques.
- $\psi \geq (n-1)^2/4$.
- However, $OPT \leq n$.

Natural IP

The natural IP formulation of the problem

IP1: max $\psi = \sum_{e \in E} y_e$, subject to

1. $y_e \leq x_u + x_v \; \forall e = (u, v) \in E$.
2. $\sum_{e \in \delta(v)} y_e \leq c_v x_v + d_v(1 - x_v) \; \forall v \in V$.
3. $x_v \in \{0, 1\} \; \forall v \in V$, $y_e \in \{0, 1\} \; \forall e \in E$.

IP1 has large Integrality Gap

- Consider K_n with uniform degree constraint $c_v = 1$.
- Set $x_v = 0.5 \; \forall \; v$. Consider disjoint cliques on $\lfloor n/2 \rfloor + 1$ and $n - \lfloor n/2 \rfloor - 1$ vertices. Set $y_e = 1$ for edges in these cliques.
- $\psi \geq (n-1)^2/4$.
- However, $OPT \leq n$.
- $\implies$ integrality gap $= \Omega(n)$.

Approximate IP

Lagrangian Like Relaxation

IP2: max $\phi = 2\sum_{e \in E} y_e - (1+\epsilon)\sum_{v \in V} z_v$, subject to

Approximate IP

Lagrangian Like Relaxation

IP2: max $\phi = 2\sum_{e\in E} y_e - (1+\epsilon)\sum_{v\in V} z_v$, subject to

1. $\sum_{e\in\delta(v)} y_e \leq c_v + z_v \ \forall v \in V$.

Approximate IP

Lagrangian Like Relaxation

IP2: max $\phi = 2\sum_{e \in E} y_e - (1+\epsilon)\sum_{v \in V} z_v$, subject to

1. $\sum_{e \in \delta(v)} y_e \leq c_v + z_v \; \forall v \in V$.
2. $z_v \in \{0, 1, 2, \ldots\} \; \forall v \in V$.

Approximate IP

Lagrangian Like Relaxation

IP2: $\max \phi = 2\sum_{e\in E} y_e - (1+\epsilon)\sum_{v\in V} z_v$, subject to

1. $\sum_{e\in\delta(v)} y_e \leq c_v + z_v \; \forall v \in V$.
2. $z_v \in \{0, 1, 2, \dots\} \; \forall v \in V$.
3. $y_e \in \{0, 1\} \; \forall e \in E$.

Approximate IP

Lagrangian Like Relaxation

IP2: max $\phi = 2\sum_{e\in E} y_e - (1+\epsilon)\sum_{v\in V} z_v$, subject to

1. $\sum_{e\in\delta(v)} y_e \leq c_v + z_v \; \forall v \in V$.
2. $z_v \in \{0, 1, 2, \ldots\} \; \forall v \in V$.
3. $y_e \in \{0, 1\} \; \forall e \in E$.

Lemma

Every maximal solution of IP2 is also a feasible solution of PDBEP.

Approximate IP

Lemma

Any α approximate solution of IP2, which is also maximal, is a $2\alpha/(1-\epsilon)$ approximation of PDBEP problem.

Approximate IP

Lemma

Any α approximate solution of IP2, which is also maximal, is a $2\alpha/(1-\epsilon)$ approximation of PDBEP problem.

Lemma

If $c_v \geq (1-\beta)d_v \, \forall v$, then any α approximate solution of IP2, which is also maximal, is a $2\alpha/(2-(1+\epsilon)\beta)$ approximation of PDBEP.

LP Relaxation

LP Relaxation of IP2

LP2: max $\phi = 2\sum_{e \in E} y_e - (1+\epsilon)\sum_{v \in V} z_v$, subject to

1. $\sum_{e \in \delta(v)} y_e \leq c_v + z_v \; \forall v \in V$.
2. $z_v \geq 0 \; \forall v \in V$.
3. $y_e \geq 0 \; \forall e \in E$.
4. $-y_e \geq -1 \; \forall e \in E$.

LP Relaxation

LP Relaxation of IP2

LP2: max $\phi = 2\sum_{e\in E} y_e - (1+\epsilon)\sum_{v\in V} z_v$, subject to

1. $\sum_{e\in\delta(v)} y_e \leq c_v + z_v \ \forall v \in V$.
2. $z_v \geq 0 \ \forall v \in V$.
3. $y_e \geq 0 \ \forall e \in E$.
4. $-y_e \geq -1 \ \forall e \in E$.

Lemma

In a corner solution of LP2 on a non-empty graph there is at least one edge e with $y_e = 0$ *or* $y_e \geq 1/2$.

Algorithm using LP2

Iterative Rounding Algorithm

- Solve LP2.

Algorithm using LP2

Iterative Rounding Algorithm

- Solve LP2.
- Discard edge(s) with $y_e = 0$.

Algorithm using LP2

Iterative Rounding Algorithm

- Solve LP2.
- Discard edge(s) with $y_e = 0$.
- Include edge(s) with $y_e \geq 1/2$ in the solution.

Algorithm using LP2

Iterative Rounding Algorithm

- Solve LP2.
- Discard edge(s) with $y_e = 0$.
- Include edge(s) with $y_e \geq 1/2$ in the solution.

Lemma

The Iterative Rounding Algorithm returns a feasible solution of PDBEP.

Algorithm using LP2

Lemma

The Iterative Rounding Algorithm gives a $1.5/(1-\epsilon)$ *approximation of IP2.*

Algorithm using LP2

Lemma

The Iterative Rounding Algorithm gives a $1.5/(1-\epsilon)$ *approximation of IP2.*

Theorem

The Iterative Rounding Algorithm approximates PDBEP with approximation factor $3/(1-\epsilon)^2$.

Algorithm using LP2

Lemma

The Iterative Rounding Algorithm gives a $1.5/(1-\epsilon)$ *approximation of IP2.*

Theorem

The Iterative Rounding Algorithm approximates PDBEP with approximation factor $3/(1-\epsilon)^2$.

Theorem

If $c_v \geq (1-\beta)d_v$ *for all* v, *then The Iterative Rounding Algorithm approximates PDBEP with approximation factor* $\frac{3}{(2-(1+\epsilon)\beta)(1-\epsilon)}$.

Conclusion

Summary

- A very simple factor 2 algorithm that takes linear time.

Conclusion

Summary

- A very simple factor 2 algorithm that takes linear time.
- A very expensive iterative rounding algorithm that beats factor 2 only for rare inputs.

Conclusion

Summary

- A very simple factor 2 algorithm that takes linear time.
- A very expensive iterative rounding algorithm that beats factor 2 only for rare inputs.

Future Directions

Conclusion

Summary

- A very simple factor 2 algorithm that takes linear time.
- A very expensive iterative rounding algorithm that beats factor 2 only for rare inputs.

Future Directions

- Establishing lower bounds for both the weighted and unweighted PDBEP.

Conclusion

Summary

- A very simple factor 2 algorithm that takes linear time.
- A very expensive iterative rounding algorithm that beats factor 2 only for rare inputs.

Future Directions

- Establishing lower bounds for both the weighted and unweighted PDBEP.
- LP based solution for the weighted case.

Conclusion

Summary

- A very simple factor 2 algorithm that takes linear time.
- A very expensive iterative rounding algorithm that beats factor 2 only for rare inputs.

Future Directions

- Establishing lower bounds for both the weighted and unweighted PDBEP.
- LP based solution for the weighted case.
- Generalization to hypergraphs.

Thanks!