CS4234 Optimisation Algorithms

Notes

 $\mathrm{AY}2024/25$ Semester 1 • Prepared by Tian Xiao @snoidetx

Deterministic	Randomized	LP + Rounding
Two special cases: ① Vertex cover on a tree; ② Known upper bound k. Deterministic Vertex Cover (2-approximation) Repeat until no remaining edge: ① Pick a rondom edge (u,v); ③ Add both u and v to vertex cover; $G \longrightarrow G_{-u-v}$.	Randomized VertexCover (2-approximation) Repeat until no remaining edge: ① Pick a random edge (u,v); ② Let z=u or v w.p. ½; ③ Add z to vertex cover; G→ G-z.	For weighted vertex coveramin $\sum_{v \in V} w_v \pi_v$ s.t. $x_n + x_v \ge 1$, $Y(u,v) \in E$ $x_v \in \{0,1\}$, $Y \in V$. $\frac{relax}{} [0,1]$ Ronnoling. If $x_v \ge \frac{1}{2}$, add v to vertex cover. (2-approximation)
Greedy Set Cover (O(logn) - approximation) Repeat nntil all elements are covered: ① Choose the set S; that covers the most uncovered elements; ② X → X\S; MST+DES (for GR-R, 2-approximation)	Linear Programming max $c^T X$ S.t. $A x \ge b$ $X \ge 0$ Simplex Method $(O(M^n))$. Think any fractible vertex v .	Feosibility is in NP \cap co-NP ① No solution \Rightarrow \exists polynomial λ ② A solution exists \Rightarrow \exists polynomial solution. Ellipsoid Method (polynomial time) $b-\varepsilon = Ax \le b+\varepsilon$, where $\varepsilon \in \frac{1}{2}$ LP Duality \Rightarrow of both fixth portional solution.
① Construct the complete graph G ② Let T be MST of G. ③ Let C be the cycle by DFS of T. Christofides Algorithm (for M, 1.5-approximation) ① T← MST of G. Add T's edges to E. ② Let 0 be nodes in T with odd dagree. 0] is even. ③ M ← min cost perfect matching for 0. ④ G← (X, EUM) (multigraph).	② VI,, VK ← neignbors of V. ③ Calculate f(v) and f(vi)f(VK). If f(v) is the max, stop and return f(v) ④ Otherwise, choose one neighbor V; S.t. f(vj) > f(v). Set v = Vj. ⑤ Return to Step ⑤. Farka's Lemma / LP Duality If AX ≥ b has no solution, then ∃ λ ≥ 0 € R™ st. λ ^T AX = 0 and λ ^T b = 1. Pandomized Weighted k - CNF - SAT). Maximum Bipartite Matching Max I will to work the matching St. Nillingte Tuv Wnv St. Nillingte Tuv = 1, Yv & V D \(\times \times \) ILP solution Semidefinite Programming relaxation Pandom hyperplane rounding = 0.87856- approximation max \(\times \) m May \(\times \) m
Greedy CNF - SAT $\left(\left(1-\frac{1}{2^{\frac{1}{2}}}\right) - \text{approximation for } k\text{- CNF-SAT};$ $\frac{\sqrt{S-1}}{2} - \text{approximation for general CNF-SAT}\right)$	For each x_i , let $x_i = 1$ or 0 w.p. $\frac{1}{2}$. Romdomized Weighted CNF-SAT $(\frac{\sqrt{S-1}}{2}$ approximation)	S.t. $\sum_{\text{tve } iji} \vec{\lambda}_i \stackrel{\Gamma}{\rightarrow} \sum_{\text{ve } iji} (-\vec{\lambda}_i) \ge y_j$, $\forall j=1,\cdots,m$ $\vec{\lambda}_i \in \{0,1\}$, $\forall i=1,\cdots,m$ $\vec{\lambda}_j \in \{0,1\}$, $\forall j=1,\cdots,m$ Pounding = $\hat{\vec{\lambda}}_i = 1$ w.p. $\vec{\vec{\lambda}}_i = 1$ $((1-\frac{1}{e}) - \text{approximation})$
Lovász Local Lemma Let o be a CSP. If 3/4c E(0,1) for all	Mayer's Algorithm (Ic the itera ① Randomly assign values to all va	ortions) aniables · Cj =
Ford-Fulkerson Algerithm O(IEIt) ① Let f(u→v) ← D, V u→v ∈ E ② While s can reach t in Gt; (a) Use BFs or DFs to find an anguering path P from s to t in 1b) F ← min cf(u→v) (c) For all u→v ∈ P, update its flow val i. f(u→v) ← f(u→v) + F. ii. f(v→u) ← f(v→u) - F. ③ Return f. Application: ① Edge-disjoint paths ② Vertex capacities ③ Bipartite matching	Flow-Decomposition Theorem Every non-negative (S,t)-flow can as a tve linear combination of dipath flows and directed cycle flow appears in a flow iff the auromat of the edge is tve # pathst#cycles Edmonds-Karp Fat Pipes Algorith (a) Let f(n - v) \(\to \) 0, \(\to \) u- (a) While s can reach t in (a) Use Dijktra's algorithm an augmenting path P from with the largost bottler in f(n - v) \(\to \) proposition for all n - v \(\to \) P, update in \(\to \) (b) For all n - v \(\to \) P, update (a) Return f. (b) Minimising the teaching stronft.	irected (S.t) ows. An edge if flow through = E . m O(E ^2 log V log f*) Dinitz Algorithm ov GE o(E ^2 V) Gt: m togolitain Use BFS to obtain oun s to t in Gt, => angmenting path neck value F. with least no of vertices F. its flow value: t F.
	Two special coses: () Vertex cover on a tree; (2) Known upper bound k. Deterministic Vertex Cover (2-approximation) Repect until no remaining edge: (1) Pick a random edge (u,v); (2) Add both u and v to vertex cover; (3) G (r-u-v) Greedy Set Cover (0 (log n) - approximation) Repeat nntil all elements are covered: (1) Choose the set S; that covers the most uncovered elements; (2) X - X \ S; MST + DFS (for Gr R, 2-approximation) (1) Construct the complete graph Gr (2) Let T be MST of Gr. (3) Let C be the cycle by DFS of T. Christofides Algorithm (for M, 1.5-approximation) (1) T - MST of Gr. Add T's edges to E. (2) Let O be nodes in T with oold dagree. (10) Is even. (3) M + min cost perfect matching for O. (4) Gr - (X, EUM) (multigraph). (5) Return Enlerian cycle C for Gr. Greedy CNF-SAT ((1-2) - approximation for k- (NF-SAT; MST-approximation for general CNF-SAT) For each Xi, choose arrandom E[W X1,, Xi]. Lovász Local Lemma Let p be a CSP. If Juc E(O.) for all cf general CNF-SAT) For each Xi, choose arrandom Let Gr = (V, E) with capality c be a flow network, and fix any sit e v. There exists a fexible flow f and a cut (S, T) on Gr St. f Saturates every edge from S to T and avoids any edge from T to S, with Iffe S, T Ford-Fulkerson Algorithm O(E f) (1) Let f(N-v) - O, V N-v E E (2) While S can reach t in Gt: (a) Use BFS or DFS to find an anymenting path P from s to t in Ib) F - min pay G P, update its flow val in fluovy - F, update its flow val in fluovy - F. (3) Return f. Application: (1) Edge-disjoint poths (2) Vertex capacities (3) Bipartite matching	Two spewol cross: ① Verter cover on a tree; ② Known upper bound k. Determinity VerterCover (2-approximation) Report until no remaining edge: ① Pick a random edge (uv)) ② Let = u or v wp. ½; ② Pokk a random edge (uv)) ③ Let = u or v wp. ½; ③ Add bith u ond v to vertex cover) ⑥ Add bith u ond v to vertex cover) ⑥ Add bith u ond v to vertex cover) ⑥ Construct the cet 5; that covers the major tuncovered clements: ③ X → X \ S. MST DES (fir Gr. R. 2-approximation) ⑥ Construct the compete group G ② Let T be MST of Gr. ③ Let C be the cycle by DFS of T. ② Let C be the cycle by DFS of T. ② Let C be modes in T with odd dagree: [O] Is even. ③ M+ with cet purplex morthing for O ⑥ Gr+ (X, EUM) (multigraph). ⑤ Return Enlerion cycle C for Gr. ② M+ with cet purplex morthing for O ⑥ Gr+ (X, EUM) (multigraph). ⑥ Return Enlerion cycle C for Gr. **Greedy(NF-SAT in Sponsington)** ⑤ Provide Xi, chose anison for general (NF-SAT in Sponsington)** ⑤ Provide Xi, chose anison for general (NF-SAT in Sponsington)** ⑤ Provide Xi, chose anison for general (NF-SAT in Sponsington)** ⑥ Provide Xi, chose anison for general (NF-SAT in Sponsington)** MFMC Theorem Let Gr-(V, E) with copulty c be a flow network, and fix any sit E v. Tree easts a fossible flow of and a cut (S,T) on first, fixer for St. (S,T) on fi