A. Approximating Submodular Set Functions with Graph Cuts

It is shown in (Devanur et al., 2013) that submodular set functions defined on a ground set of n elements can be $O\left(n^2\right)$ approximated by directed graph cuts. We state this fact as a lemma, and we include the proof for completeness in Section A.1 below.

Definition A.1. *Given a submodular set function $F : 2^V \to \mathbb{R}$, such that $F(\emptyset) = F(V) = 0$, and a weighted directed graph $G = (V, E, c)$, we say that the cut function of G α-approximates F if*

$$\frac{1}{\alpha} c^+(A) \leq F(A) \leq c^+(A)\,, \text{for all } A \subseteq V\,.$$

Algorithm 2 Approximate non-negative submodular function $F = F^0 - w^0$ by graph cuts, where $F^0 : 2^V \to \mathbb{R}_{\geq 0}$ is the initial submodular function and the shift vector $w^0 : V \to \mathbb{R}$ is given as input

```
function GraphApprox(w^0 : V → ℝ)
Call GraphApproxShifted(F^0 − w^0)
function GraphApproxShifted(F : 2^V → ℝ_{≥0})
Let E = {(u, v) ∈ V × V : u ≠ v}.
for u, v ∈ V : u ≠ v do
    Compute w_uv = min_{A⊆V: u∈A, v∉A} F(A).
    c_uv = w_uv .
end for
return G = (V, E, c)
```

Lemma A.2. *Let $V = \{1, \ldots, n\}$, and let $F : V \to \mathbb{R}$ be a non-negative submodular set function, satisfying $F(\emptyset) = F(V) = 0$. Using $O\left(n^2\right)$ calls to a minimization oracle which can compute for all pairs $u, v \in V$*

$$\min_{\substack{A \subseteq V \\ u \in A, v \notin A}} F\left(A\right)$$

one can compute a weighted directed graph $G\left(V, E, c\right)$ such that its cut function

$$c^+\left(A\right) := \sum_{\substack{(u,v) \in E: \\ u \in A, v \notin A}} c_{uv}$$

$(n^2/4)$-approximates G. In other words, for any $A \subseteq V$ the size of the graph cut satisfies:

$$\frac{1}{n^2/4} \cdot c^+\left(A\right) \leq F\left(A\right) \leq c^+\left(A\right)\ .$$

Furthermore, if F takes only values that are discrete multiples of Δ, i.e. $F(A) \in \Delta \cdot \mathbb{Z}_{\geq 0}$ for all A, then all elements of c are discrete multiples of Δ.

As a consequence, we obtain a good approximation by graph cuts for decomposable submodular functions where each component in the decomposition acts on few elements, i.e., when $F_i(A) = F_i(A \cap V_i)$ for some $V_i \subseteq V$.

Lemma A.3. *Let $V = \{1, \ldots, n\}$, and let $F_i : V_i \to \mathbb{R}$, $V_i \subseteq V$ be non-negative submodular set functions, with $F_i(\emptyset) = F_i(V_i)$, for $i = 1, \ldots, r$. In the time required to compute for all pairs $u \neq v \in V$ and for all $1 \leq i \leq r$*

$$\min_{\substack{A \subseteq V_i \\ u \in A, v \notin A}} F_i\left(A\right)$$

one can compute a weighted directed graph $G\left(V, E, c\right)$ such that its cut function $(M^2/4)$-approximates $\sum_{i=1}^r F_i$, where $M = \max_{i=1,r} |V_i|$.

Proof. For each i compute the corresponding graph as in Lemma A.2. Then take the union of edges over the same vertex set. □

We showed that the function $F(A)$ is well approximated by the cut function $c^+(A)$ for the graph we constructed. Note that c^+ is only defined on internal vertices of the graph, excluding s and t. However this does not affect its submodularity. Therefore the submodular base polytopes for the two function approximate each other well.

Lemma A.4. *Let F, G be two submodular functions defined over the same vertex set V such that $F(\emptyset) = G(\emptyset) = 0$, $F(V) = G(V) = 0$, and for any $A \subseteq V$, $\frac{1}{\alpha} G(A) \leq F(A) \leq G(A)$. Then their submodular base polytopes satisfy:*

$$\frac{1}{\alpha} B(G) \subseteq B(F) \subseteq B(G) .$$

Proof. Let any $w \in B(F)$. Then $w(V) = F(V) = G(V)$. Furthermore for any set $A \subseteq V$, we have $w(A) \leq F(A) \leq G(A)$. Similarly for any $w \in G(A)$, we have $w(A) \leq G(A) \leq \alpha F(A)$, so $B(G) \subseteq \alpha B(F)$, which yields the claim. □

At this point we can prove that the submodular base polytope of the cut function created in Lemma A.3 approximates the submodular base polytope of the decomposable function $\sum_{i=1}^{r} F_i$.

Lemma A.5. *Let $V = \{1, \ldots, n\}$, let $F_i : V_i \to \mathbb{R}$, $V_i \subseteq V$ be non-negative submodular set functions, with $F_i(\emptyset) = F_i(V_i)$, for $i = 1, \ldots, r$, and let $F = \sum_{i=1}^{r} F_i$. In the time required to solve $\min_{A \subseteq V_i : u \in A, v \notin A} F_i(A)$ for all $u, v \in V_i$ and all i, we can compute a weighted directed graph $G = (V, E, c)$ such that the submodular base polytope of the cut function $c^+(A)$ satisfies $\frac{1}{M^2/4} B(c^+) \subseteq B(F) \subseteq B(c^+)$, where $M = \max_{i=1,r} |V_i|$.*

Proof. The proof follows directly from applying Lemma A.3, followed by Lemma A.4. □

A.1. Proof of Lemma A.2

Proof. To simplify notation let us denote by

$$w_{uv} = \min_{\substack{A \subseteq V \\ u \in A, v \notin A}} F(A) \ ,$$

and let T_{uv} be the set achieving this minimum.

Consider the graph defined as follows. For every $u, v \in V$, create an arc (u, v) with weight $c_{uv} = w_{uv}$. By construction all capacities are discrete multiples of Δ.

Now we can prove the lower bound on F. We have that

$$c^+(A) = \sum_{u \in A, v \notin A} c_{uv} \leq \sum_{u \in A, v \notin A} c_{uv} = \sum_{u \in A, v \notin A} F(T_{uv}) \leq \sum_{u \in A, v \notin A} F(A) \leq \left(n^2/4\right) F(A) .$$

We used the fact that $F(A)$ upper bounds c_{uv} for all $u \in A, v \notin A$. Now we prove the upper bound. For any nonempty set $A \subset V$ we can write $A = \bigcup_{u \in A} \left(\bigcap_{v \in V \setminus A} T_{uv} \right)$. By twice applying Lemma A.6, we obtain that

$$F(A) \leq \sum_{u \in A} \sum_{v \notin A} F(T_{uv}) = c^+(A) \ .$$

Additionally, we have by construction that

$$c^+(\emptyset) = c^+(V) = 0 .$$

□

Lemma A.6. *Let F be a non-negative submodular set function $F : 2^V \to \mathbb{R}$, an let $A_1, \ldots, A_t$ be subsets of V. Then*

$$F\left(\bigcup_{i=1}^{t} A_i\right) \leq \sum_{i=1}^{t} F(A_i)$$

and

$$F\left(\bigcap_{i=1}^{t} A_i\right) \leq \sum_{i=1}^{t} F(A_i) .$$

Proof. We prove by induction on t. If $t = 1$, both inequalities are equalities. Otherwise, suppose they hold for $t-1$. Let $S = \bigcup_{i=1}^{t-1} A_i$. By submodularity, $F(S \cup A_t) \le F(S) + F(A_t) - F(S \cap A_t)$. Since F is non-negative, so is $F(S \cap A_t)$, and therefore $F(S \cup A_t) \le F(S) + F(A_t)$. Applying the induction hypothesis this concludes the first part of the proof.

Similarly, let $S = \bigcap_{i=1}^{t-1} A_i$. By submodularity, $F(S \cap A_t) \le F(S) + F(A_t) - F(S \cup A_t)$. Since F is non-negative, so is $F(S \cup A_t)$, and therefore $F(S \cap A_t) \le F(S) + F(A_t)$. Again, applying the induction hypothesis this concludes the second part of the proof. □

B. Parametric Submodular Minimization via Optimization on the Base Polytope

In this section, for completeness, we provide a proof of Lemma 2.4, which is based on (Bach, 2011) (see Chapter 8). In addition, we provide error analysis for reductions between approximate solutions to the combinatorial parametric submodular minimization problem, its continuous version involving the Lovász extension, and the dual formulation on the base polytope.

Proof of Lemma 2.4. Given any point x, let $\beta \le \min\{0, \min_i x_i\}$. Applying the definition of the Lovász extension, and the fundamental theorem of calculus, we can write:

$$\begin{aligned} f(x) + \sum_{i \in V} \psi_i(x_i) &= \int_0^\infty F(\{i : x_i \ge t\}) dt + \int_\beta^0 \left(F(\{i : x_i \ge t\}) - F(V)\right) dt \\ &\quad + \sum_{i \in V} \psi_i(\beta) + \int_\beta^\infty \sum_{i : x_i \ge t} \psi_i'(t) dt \\ &= \int_\beta^\infty \left(F(\{i : x_i \ge t\}) + \sum_{i : x_i \ge t} \psi_i'(t) \right) dt + \sum_{i \in V} \psi_i(\beta) - \beta F(V) \,. \end{aligned}$$

Note that we crucially used the fact that the parametric term $\sum_i \psi_i'(t)$ is separable.

Next we show that if the optimal sets A^α were different from those defined in (3), then we could obtain a different iterate x' such that $f(x') + \sum_{i \in V} \psi_i(x') \le f(x^*) + \sum_{i \in V} \psi_i(x^*)$. However, since ψ is strictly convex, the minimizer of $f(x) + \sum_{i \in V} \psi_i(x)$ is unique. This gives a contradiction leading us to the desired conclusion.

Indeed, let $x_i' = \sup_{i \in A^\alpha} \alpha$. By the strict convexity property of ψ_i, we have that for any $\alpha > \beta$, $A^\alpha \subseteq A^\beta$, which we reprove for completeness in Lemma B.1.

Using this fact, we know that if A^α are the optimizers of $F(A) + \sum_{i \in V} \psi_i(\alpha)$, then we can write:

$$\int_\beta^\infty \left(F(A^t) + \sum_{i \in A^t} \psi_i'(t) \right) dt = \int_\beta^\infty \left(F(\{i : x_i' \ge t\}) + \sum_{i : x_i' \ge t} \psi_i'(t) \right) dt \,.$$

Since by the optimality of A^t we have that

$$f(A^t) + \sum_{i \in A_t} \psi_i(t) \le F(\{i : x_i \ge t\}) + \sum_{i : x_i \ge t} \psi_i'(t) \,,$$

it means that letting $\beta = \min\{0, \min_i x_i', \min_i x_i^*\}$,

$$\int_\beta^\infty \left(F(\{i : x_i' \ge t\}) + \sum_{i : x_i' \ge t} \psi_i'(t) \right) dt \le \int_\beta^\infty \left(F(\{i : x_i^* \ge t\}) + \sum_{i : x_i^* \ge t} \psi_i'(t) \right) dt$$

and therefore

$$f(x') + \sum_{i \in V} \psi_i(x') \le f(x^*) + \sum_{i \in V} \psi_i(x^*) \,,$$

which concludes the proof. □

Lemma B.1. *Let $F : 2^V \to \mathbb{R}$ be a submodular set function, and let $\psi_i : \mathbb{R} \to \mathbb{R}$ be a family of strictly convex functions, for $i \in V$. Let $F_\alpha(A) = F(A) + \sum_{i \in A} \psi_i'(\alpha)$, and $A^\alpha = \arg\min_{A \subseteq V} F_\alpha(A)$. If $\alpha > \beta$, then $A^\alpha \subseteq A^\beta$.*

Proof. By optimality we have that

$$F(A^\alpha) + \sum_{i \in A^\alpha} \psi_i'(\alpha) \le F(A^\alpha \cap A^\beta) + \sum_{i \in A^\alpha \cap A^\beta} \psi_i'(\alpha)$$

and

$$F(A^\beta) + \sum_{i \in A^\beta} \psi_i'(\beta) \le F(A^\alpha \cup A^\beta) + \sum_{i \in A^\alpha \cup A^\beta} \psi_i'(\beta)\,.$$

Summing up we obtain that

$$\begin{aligned}
&\sum_{i \in A^\alpha} \psi_i'(\alpha) - \sum_{i \in A^\alpha \cap A^\beta} \psi_i'(\alpha) + \sum_{i \in A^\beta} \psi_i'(\beta) - \sum_{i \in A^\alpha \cup A^\beta} \psi_i'(\beta) \\
&\le F(A^\alpha \cap A^\beta) + F(A^\alpha \cup A^\beta) - F(A^\alpha) - F(A^\beta) \\
&\le 0\,,
\end{aligned}$$

where we used submodularity in the last step. Therefore

$$\sum_{i \in A^\alpha \setminus A^\beta} (\psi_i'(\alpha) - \psi_i'(\beta)) \le 0\,.$$

Hence we conclude that $A^\alpha \setminus A^\beta = \emptyset$, since by strict convexity we have that for all i, $\psi_i'(\alpha) > \psi_i'(\beta)$, which would make the term above strictly positive had there been any elements in the set difference.

□

Next we perform a careful error analysis to bound the total error we incur in the case where the iterate we consider is not an exact minimizer of (2), but has some small error in norm.

Lemma B.2. *Under the conditions from Lemma 2.4, let $\widetilde{x} \in \mathbb{R}$ be a point satisfying $\|\widetilde{x} - x^*\| \le \epsilon$, where x^* is the minimizer of (2). Let the sets*

$$\widetilde{A}^\alpha = \{i : \widetilde{x}_i \ge \alpha\}\,.$$

If ψ_i is σ-strongly convex, for all i, and $\max_{A \subseteq V} F(A) - \min_{A' \subseteq V} F(A') \le M$, then:

$$F(\widetilde{A}^\alpha) + \sum_{i \in \widetilde{A}^\alpha} \psi_i'(\alpha) \le F(A^\alpha) + \sum_{i \in A^\alpha} \psi_i'(\alpha) + Mn^{3/2}\epsilon + \beta\epsilon^2/2\,.$$

Proof. First, using the smoothness of ψ_i we prove that

$$f(\widetilde{x}) + \sum_i \psi_i(\widetilde{x}_i) \le f(x^*) + \sum_i \psi_i(x_i^*) + Mn^{3/2}\epsilon + \beta\epsilon^2/2\,.$$

To prove this we first note that f is Lipschitz, since we can use the fact that entries of the gradient of the Lovász extension consist of differences between F evaluated at different subsets of V. Hence for any x, $|\nabla_i f(x)| \le \max_{A \subseteq V} F(A) - \min_{A' \subseteq V} F(A') \le M$, and thus $\|\nabla f(x)\| \le M\sqrt{n}$. Therefore

$$f(\widetilde{x}) - f(x^*) \le M\sqrt{n}\|\widetilde{x} - x^*\| \le M\sqrt{n}\epsilon\,.$$

Secondly, we use the smoothness of ψ_i, to obtain that

$$\psi_i(\widetilde{x}_i) \le \psi_i(x_i^*) + \psi_i'(x_i^*)(\widetilde{x}_i - x_i^*) + \frac{\beta}{2}(\widetilde{x}_i - x_i^*)^2\,.$$

Using Lemma B.5 we see that $\psi_i'(x_i^*) = -w_i^*$, where w^* is the optimizer of a certain function over the base polytope $B(F)$. By the definition of $B(F)$ we have $w_i^* \le F(\{i\}) \le M$ and $-w_i^* + \sum_{j\neq i} w_j^* = F(V)$, so $-w_i^* \ge F(V) - \sum_{j\neq i} F(\{j\}) \ge -M(n-1)$. Thus, by applying Cauchy-Schwarz, we have

$$\sum_{i\in V} \psi_i'(x_i^*)(\widetilde{x}_i - x_i^*) \le \max_i |\psi_i'(x_i^*)|\sqrt{n} \cdot \|\widetilde{x} - x^*\| \le M(n-1)n^{1/2}\epsilon\,,$$

and thus

$$\sum_{i\in V} \psi_i'(\widetilde{x}_i) - \psi_i'(x_i^*) \le M(n-1)n^{1/2}\epsilon + \beta\epsilon^2/2\,.$$

Combining with the bound on $f(\widetilde{x})$, we obtain our claimed error in function value.

Now we can finalize the argument. Following the proof of Lemma 2.4 we write $f(\widetilde{x}) + \sum_{i\in V} \psi_i'(\widetilde{x}_i)$ as an integral, and similarly for x^*, to conclude that for $\beta = \min\{0, \min_i \widetilde{x}_i, \min_i x_i^*\}$,

$$\int_\beta^\infty \left(F(\widetilde{A}^t) + \sum_{i\in\widetilde{A}^t} \psi_i'(t) \right) dt \le \int_\beta^\infty \left(F(A^t) + \sum_{i\in A^t} \psi_i'(t) \right) dt + Mn^{3/2}\epsilon + \beta\epsilon^2/2\,.$$

Since by definition A^t minimizes $\sum_{i\in A^t} \psi_i'(t)$, we conclude that for all t,

$$F(\widetilde{A}^t) + \sum_{i\in\widetilde{A}^t} \psi_i'(t) \le F(A^t) + \sum_{i\in A^t} \psi_i'(t) + Mn^{3/2}\epsilon + \beta\epsilon^2/2\,.$$

□

We can also show that if we obtain an approximate minimizer of the dual problem (4) over $B(F)$, we can use it to recover an approximate minimizer of the primal problem (2).

Lemma B.3. *Let w^* be the minimizer of the dual problem (4), and and let x^* be the minimizer of the primal problem (2). If $w \in B(F)$ such that*

$$\sum_{i\in V} \psi_i^*(-w_i) \le \sum_{i\in V} \psi_i^*(-w_i^*) + \epsilon\,,$$

then the point $x \in \mathbb{R}^n$ where $x_i = (\psi_i^*)'(-w_i)$ satisfies

$$\|x - x^*\| \le \sqrt{\frac{2L\epsilon}{\sigma^2}}\,.$$

Proof. By Lemma B.5 we know that x^* and w^* are related via $x_i^* = (\psi_i^*)'(-w_i^*)$. Therefore we can write

$$|x_i - x_i^*| = |(\psi_i^*)'(-w_i) - (\psi_i^*)'(-w_i^*)| \le \frac{1}{\sigma}|w_i - w_i^*|\,,$$

where in the last inequality we used the fact that ψ_i is σ-strongly convex, and hence ψ_i^* is $1/\sigma$-smooth (Shalev-Shwartz & Singer, 2006; Kakade et al., 2012). Next we show that $|w_i - w_i^*|$ is bounded by a function of ϵ.

Since by assumption ψ_i is L-smooth, its dual ψ_i^* is $1/L$-strongly convex. Therefore we have that, for all i:

$$\psi_i^*(-w_i) \ge \psi_i^*(-w_i^*) + (\psi_i^*)'(-w_i^*) \cdot (-w_i - (-w_i^*)) + \frac{\sigma}{2}(w_i^* - w_i)^2\,.$$

Furthermore, since w^* is an optimizer over $B(F)$, we know by first-order optimality that for any $w \in B(F)$:

$$\sum_{i\in V} (\psi_i^*)'(-w_i^*) \cdot (-w_i - (-w_i^*)) \ge 0\,,$$

i.e. slightly moving the point from $-w^*$ towards $-w$ can only increase function value. Thus we obtain that

$$\sum_{i\in V} \psi_i^*(-w_i) \ge \sum_{i\in V} \psi_i^*(-w_i^*) + \frac{1}{2L} \sum_{i\in V} (w_i^* - w_i)^2\,.$$

Combining with the hypothesis, this implies that

$$\frac{1}{2L}\sum_{i\in V}(w_i^* - w_i)^2 \leq \epsilon\,,$$

and therefore

$$\|x - x^*\|^2 \leq \frac{1}{\sigma^2}\sum_{i\in V}(w_i - w_i^*)^2 \leq \frac{2L\epsilon}{\sigma^2}\,,$$

which implies the claimed result. □

As a corollary of the previous lemmas, we see that an approximate solution to the dual problem (4) yields an approximate solution to the original parametric problem (1).

Corollary B.4. *Let $F : 2^V \to \mathbb{R}$ be a non-negative submodular set function, and let the the family of parametric problems defined in (1). Let $w \in B(F)$ such that*

$$\sum_{i\in V}\psi_i^*(-w_i) \leq \sum_{i\in V}\psi_i^*(-w_i^*) + \epsilon\,,$$

where w^* is the true minimizer of the dual problem (4). Then for any α, the set

$$\widetilde{A}^\alpha = \{i : \psi_i^*(-w_i) \geq \alpha\}$$

satisfies

$$F_\alpha(\widetilde{A}^\alpha) \leq F_\alpha(A^\alpha) + \sqrt{\epsilon}\cdot Mn^{3/2}\sqrt{2L/\sigma^2} + \epsilon\cdot(L/\sigma)^2\,,$$

where $A^\alpha = \arg\min_{A\subseteq V} F_\alpha(A)$.

Proof. From Lemma B.3 we know that the hypothesis implies that the point x where $x_i = (\psi_i^*)(-w_i)$ satisfies $\|x - x^*\| \leq \sqrt{2L\epsilon/\sigma^2}$. Applying Lemma B.2 we thus obtain that the sets constructed satisfy

$$F_\alpha(\widetilde{A}^\alpha) \leq F_\alpha(A^\alpha) + Mn^{3/2}\sqrt{2L\epsilon/\sigma^2} + L/2\cdot(2L\epsilon/\sigma^2)\,,$$

which yields our claim. □

The following helper lemma shows that we can efficiently convert between (exact) solutions to the primal and dual problems (2) and (4). Using standard techniques we can prove that these also enable us to convert between suboptimal solutions, while satisfying certain error bounds.

Lemma B.5. *Let x^* be the (unique) minimizer of (2), and let w^* be the minimizer of (4). Then $w_i^* = -\psi_i'(x_i)$ and $(\psi_i^*)'(-w_i) = x_i$, for all $i \in V$.*

Proof. We the dual characterization of f and Sion's theorem, to write

$$\min_{x\in\mathbb{R}^n} f(x) + \sum_{i\in V}\psi_i(x_i) = \min_{x\in\mathbb{R}^n}\max_{w\in B(F)}\langle w, x\rangle + \sum_{i\in V}\psi_i(x_i) = \max_{w\in B(F)}\min_{x\in\mathbb{R}^n}\langle w, x\rangle + \sum_{i\in V}\psi_i(x_i)\,.$$

Since each ψ_i acts on a different coordinate we can write the inner minimization problem as

$$\min_{x\in\mathbb{R}^n}\sum_{i\in V}(w_i x_i + \psi_i(x_i)) = -\sum_{i\in V}\psi_i^*(-w_i)\,,$$

where we applied the definition of the Fenchel dual. Furthermore by standard convex analysis (Borwein & Lewis, 2010; Rockafellar, 1970), as ψ_i' ranges from $-\infty$ to ∞ for each i we have that $(\psi_i^*)'(-w_i) = x_i$, and similarly $\psi_i'(x_i) = -w_i$.

Thus we can equivalently write (2) as

$$\max_{w\in B(F)} -\sum_{i\in V}\psi_i^*(-w_i)\,.$$

By the previous observation, the optima are thus related via $(\psi_i^*)'(-w_i^*) = x_i^*$, and similarly $\psi_i'(x_i^*) = -w_i^*$. □

C. Parametric s-t Cuts

In this section we show how to solve the parametric minimum cut problem by efficiently using a maximum flow oracle. In Section 4 we show how to convert the solution obtained by this combinatorial routine to a nearly-optimal solution to a related optimization problem on the submodular base polytope of the corresponding cut function.

In the parametric min s,t-cut problem, we are given a directed network $G=(V,E)$ with two distinguished vertices: a *source* $s\in V$, and a *sink* $t\in V$, $s\neq t$. The capacities of individual edges of G are nonnegative functions of a real parameter λ in some possibly infinite domain $\mathbb{D}\subseteq\mathbb{R}$ (as opposed to constants in the classical setting of min s,t-cut). Following (Gallo et al., 1989), we assume that the capacities of edges $sv\in E$ are nondecreasing in λ and the capacities of edges $vt\in E$ are nonincreasing in λ. The capacities of all other edges of G are constant.

We denote by $c_\lambda(uv):\mathbb{D}\to\mathbb{R}$ the capacity function of an edge $uv\in E$. Moreover, we assume that these edge capacity functions can be evaluated for arbitrary λ in constant time.

Roughly speaking, the goal of the parametric min s,t-cut problems is to compute a representation of min s,t-cut for all the possible parameters λ. Before we precisely define what this means, let us introduce some more notation and state some useful properties of (parametric) min-cuts.

Denote by $\text{cap}(G)$ the capacity of a min s,t-cut in G. Let $G_{\lambda'}$ be the graph with all the parameterized capacities replaced with the corresponding values for $\lambda=\lambda'$. For any S, $s\in S\subseteq V\setminus\{t\}$, let $c_\lambda(S)$ be the capacity function of S, i.e., the sum of capacity functions $c_\lambda(uv)$ through all uv with $u\in S$ and $v\in V\setminus S$.

Lemma C.1 ((Ford & Fulkerson, 1962)). *For any G, there exists a unique* minimal *minimum s,t-cut (S,T) with $|S|$ smallest possible, such that for any min s,t-cut (S',T') of G we have $S\subseteq S'$. Given any maximum s,t-flow f in G, such a cut can be computed from f in $O(m)$ time.*

Proof. Let G_f be the residual network associated with flow f. We let S be the set of vertices reachable from s in G_f (via edges with positive capacity). As proven by Ford & Fulkerson (1962, Theorem 5.5), S defined this way does not depend on the chosen maximum flow f, and $S\subseteq S'$ holds. Clearly, given f, S can be found using any graph search algorithm. □

Ford & Fulkerson (1962, Corollary 5.4) showed that for any two min s,t-cuts $(S_1,T_1),(S_2,T_2)$ of G, $(S_1\cap S_2,T_1\cup T_2)$ is also a min s,t-cut of G. Gallo et al. (1989, Lemma 2.8) gave the following generalization of this property to parametric min s,t-cuts.

Lemma C.2 ((Gallo et al., 1989)). *Let $\lambda_1\leq\lambda_2$. For $i=1,2$, let $(S_{\lambda_i},T_{\lambda_i})$ be some min s,t-cut in G_{λ_i}. Then $(S_{\lambda_1}\cap S_{\lambda_2},T_{\lambda_1}\cup T_{\lambda_2})$ is a min s,t-cut in G_{λ_1}.*

Our algorithm will use the following crucial property of parametric minimal min s,t-cuts.

Lemma C.3. *Let $\lambda_1\leq\lambda_2$. For $i=1,2$, let $(S_{\lambda_i},T_{\lambda_i})$ be the unique minimal min s,t-cut in G_{λ_i}. Then $S_{\lambda_1}\subseteq S_{\lambda_2}$.*

Proof. The uniqueness of S_{λ_1} and S_{λ_2} follows by Lemma C.1 applied to G_{λ_1} and G_{λ_2}, respectively. By Lemma C.2, $(S_{\lambda_1}\cap S_{\lambda_2},T_{\lambda_1}\cup T_{\lambda_2})$ is a min s,t-cut in G_{λ_1}. By Lemma C.1, we have $S_{\lambda_1}\subseteq S_{\lambda_1}\cap S_{\lambda_2}$. It follows that $S_{\lambda_1}\subseteq S_{\lambda_2}$. □

Now, given Lemma C.3, we can formally state our goal in this section, which is to compute a parametric min s,t-cut defined as follows. Let $\lambda_{\min}\in\mathbb{D}$ be such that the minimal min s,t-cuts of $G_{\lambda_{\min}}$ and $G_{\lambda'}$ are equal for all $\lambda'\in\mathbb{D}$, $\lambda'<\lambda_{\min}$. Similarly, let $\lambda_{\max}\in\mathbb{D}$ be such that the minimal min s,t-cuts of $G_{\lambda_{\max}}$ and $G_{\lambda'}$ are equal for all $\lambda'\in\mathbb{D}$ with $\lambda'>\lambda_{\max}$. We will consider $\lambda_{\min}$ and $\lambda_{\max}$ additional inputs to our problem.

For simplicity, in the remaining part of this section we denote by S_λ and T_λ the s-side and the t-side (resp.) of *the minimal* min-s,t-cut of G_λ.

Definition C.4 (Parametric min s,t-cut). *Let $\Lambda=\{\lambda_1,\ldots,\lambda_k\}\subseteq\mathbb{D}$, where $k\leq n-1$ and $\lambda_{\min}<\lambda_1<\ldots<\lambda_k\leq\lambda_{\max}$. Let $\lambda_0=\lambda_{\min}$. Let $\tau:V\to\Lambda\cup\{\lambda_{\min},\infty\}$ be such that $\tau(s)=\lambda_{\min}$ and $\tau(t)=\infty$. Let $S(z)=\{v\in V:\tau(v)\leq z\}$. A pair (Λ,τ) is a parametric min s,t-cut of G if:*

1. *For $i=0,\ldots,k-1$, $S(\lambda_i)$ is a minimal min s,t-cut of $G_{\lambda'}$ for all $\lambda'\in[\lambda_i,\lambda_{i+1})\cap\mathbb{D}$.*
2. *$S(\lambda_k)$ is a minimal min s,t-cut of $G_{\lambda_{\max}}$.*

3. *For $i = 0, \ldots, k-1$, $S(\lambda_i) \subsetneq S(\lambda_{i+1})$.*

It will also prove useful to define an approximate version of parametric min s,t-cut.

Definition C.5 (ε-approximate parametric min s,t-cut)**.** *Let Λ, τ, and $S : \mathbb{D} \to 2^V$ be as in Definition C.4. A pair (Λ, τ) is called an ε-approximate parametric min s,t-cut of G if:*

1. *For $i = 0, \ldots, k-1$, $S(\lambda_i)$ is a minimal min s,t-cut of $G_{\lambda'}$ for all $\lambda' \in [\lambda_i, \lambda_{i+1} - \varepsilon) \cap \mathbb{D}$.*
2. *$S(\lambda_k)$ is a minimal min s,t-cut of $G_{\lambda_{\max}}$.*
3. *For $i = 0, \ldots, k-1$, $S(\lambda_i) \subsetneq S(\lambda_{i+1})$.*

Lemma C.6. *Let (Λ, τ) be the parametric min s,t-cut of G. Let $(\Lambda_\varepsilon, \tau_\varepsilon)$ be an ε-approximate parametric min s,t-cut of G. Then for all $v \in V$, $\tau(v) \le \tau_\varepsilon(v) \le \tau(v) + \varepsilon$.*

Proof. Let $S(z) = \{v \in V : \tau(v) \le z\}$, and $S_\varepsilon(z) = \{v \in V : \tau_\varepsilon(v) \le z\}$. First of all, $\tau(v) = \infty$ if and only if $\tau_\varepsilon(v) = \infty$. This is because each of those is equivalent to $v \notin S_{\lambda_{\max}}$. In this case the lemma holds trivially.

So in the following let us assume that $\tau(v)$ and $\tau_\varepsilon(v)$ are both finite. We first prove $\tau_\varepsilon(v) \ge \tau(v)$. If $\tau(v) = \lambda_{\min}$ then this follows by $\tau_\varepsilon(v) \ge \lambda_{\min}$. So suppose $\tau(v) = \lambda$ for some $\lambda \in \Lambda$. Then by item (1) of Definition C.4, for any $\lambda' < \lambda$, $S(\lambda')$ is a minimal min s,t-cut of $G_{\lambda'}$ and $v \notin S(\lambda')$. If we had $\tau_\varepsilon(v) < \tau(v)$, then $S_\varepsilon(\tau_\varepsilon(v))$ would be a minimal min s,t-cut of $G_{\tau_\varepsilon(v)}$ such that $v \in S_{\tau_\varepsilon(v)}$ and $\tau_\varepsilon(v) < \lambda$, a contradiction.

Now let us prove $\tau_\varepsilon(v) \le \tau(v) + \varepsilon$. To this end, suppose $\tau_\varepsilon(v) > \tau(v) + \varepsilon$. If $\tau_\varepsilon(v) = \lambda_{\min}$, then we have $\lambda_{\min} > \tau(v) + \varepsilon \ge \lambda_{\min} + \varepsilon$, a clear contradiction. So let us assume that $\tau_\varepsilon(v) \in \Lambda_\varepsilon$ and let λ^* be the element preceding $\tau_\varepsilon(v)$ in Λ_ε, or $\lambda^* = \lambda_{\min}$ if no such element exists. We have $v \notin S_{\lambda^*}$ and S_{λ^*} is a minimal min s,t-cut in $G_{\lambda'}$ for $\lambda' = \lambda^*$ and all $\lambda' \in [\lambda^*, \tau_\varepsilon(v) - \varepsilon)$. As a result, for any $\lambda'' < \tau_\varepsilon(v) - \varepsilon$, the minimal min s,t-cut of $G_{\lambda''}$ does not contain v in the s-side. But $\tau(v) < \tau_\varepsilon(v) - \varepsilon$, $v \in S(\tau(v))$, and $S(\tau(v))$ is a minimal min s,t-cut of $G_{\tau(v)}$, a contradiction. ☐

Our main result in this section is the following theorem.

Theorem 3.2. *Let $R = \lambda_{\max} - \lambda_{\min}$ be an integral multiple of $\varepsilon > 0$. Let $T_{\text{maxflow}}(n', m') = \Omega(m' + n')$ be a convex function bounding the time needed to compute maximum flow in a graph with n' vertices and m' edges obtained from G_λ by edge/vertex deletions and/or edge contractions (with merging parallel edges by summing their capacities) for any $\lambda = \lambda_{\min} + \ell\varepsilon$ and any integer $\ell \in [0, R/\varepsilon]$. Then, ε-approximate parametric min s,t-cut in G can be computed in $O(T_{\text{maxflow}}(n, m \log n) \cdot \log \frac{R}{\varepsilon} \cdot \log n)$ time.*

The rest of this section is devoted to proving Theorem 3.2. For a connected subset $X \subseteq V(G)$, $\{s,t\} \not\subseteq X$, let G/X denote G after merging the vertex set X into a single vertex. If the contracted vertex set X contains s (t), then the resulting vertex inherits the identity of s (t, resp.).

Lemma C.7. *Let λ be arbitrary and let (S_λ, T_λ) be the minimal min s,t-cut in G_λ. Then:*

1. *For any $\lambda' \ge \lambda$, cap$(G_{\lambda'}) = $ cap$(G_{\lambda'}/S_\lambda)$.*
2. *For any $\lambda' \le \lambda$, cap$(G_{\lambda'}) = $ cap$(G_{\lambda'}/T_\lambda)$.*

Proof. We only prove item 1, as item 2 is analogous. Since merging vertices is equivalent to connecting them with infinite capacity edges, it cannot decrease the min s,t-cut capacity, i.e., cap$(G_{\lambda'}) \le$ cap$(G_{\lambda'}/S_\lambda)$. On the other hand, by Lemma C.3, the minimal s,t min-cut $(S_{\lambda'}, T_{\lambda'})$ in $G_{\lambda'}$ satisfies $S_\lambda \subseteq S_{\lambda'}$. Hence, the capacity of the s,t-cut $(S_{\lambda'}/S_\lambda, T_{\lambda'})$ in $G_{\lambda'}/S_\lambda$ is the same as the capacity cap$(G_{\lambda'})$ of $(S_{\lambda'}, T_{\lambda'})$ in $G_{\lambda'}$. Consequently, cap$(G_{\lambda'}) \ge$ cap$(G_{\lambda'}/S_\lambda)$. ☐

Remark C.8. *If (S_λ, T_λ) is a minimal min s,t-cut in G_λ, then $G[S_\lambda]$ is connected by construction (Lemma C.1). However, $G[T_\lambda]$ might in general consist of several connected components if G_λ contains zero-capacity edges. In that case, we can still obtain $G_{\lambda'}/T_\lambda$ above using edge/vertex deletions and edge contractions. Namely, we contract only the connected component A of T_λ that contains t. For any other component C_i ($i = 1, \ldots, q$) of T_λ, its incoming edges start in S_λ and all have capacity 0 in G_λ, and thus also in $G_{\lambda'}$ for $\lambda' < \lambda$. Consequently, removing the vertices of $\bigcup_{i=1}^q C_i$ and subsequently contracting A has the same effect on $G_{\lambda'}$ as merging the entire T_λ, i.e., $G_{\lambda'}/T_\lambda = G_{\lambda'}[V \setminus \bigcup_{i=1}^q C_i]/A$.*

We use a recursive "divide-and-conquer" algorithm. The input to a recursive procedure ApxParametricMinCut is a graph $G = (V, E)$ with n vertices, m edges, source s and sink t, the parametric capacity function $c_\lambda : E \to \mathbb{D} \to \mathbb{R}$, and two parameters $\lambda_{\min}, \lambda_{\max}$ such that ε evenly divides $\lambda_{\max} - \lambda_{\min}$. The output of the procedure is an ε-approximate parametric min s,t-cut $(\{\lambda_1, \ldots, \lambda_k\}, \tau)$ as in Definition C.5. By Lemma C.3, $k \le |V(G)| - 1$.

Algorithm 3 Computing an ε-approximate parametric min s,t-cut.

Let s, t, ε be globally defined.

function ApxParametricMinCut($G = (V, E), c_\lambda : E \to \mathbb{D} \to \mathbb{R}, \lambda_{\min} \in \mathbb{D}, \lambda_{\max} \in \mathbb{D}$)
if $|V| \le 2$ **then**
  return $(\emptyset, \{s \to \lambda_{\min}, t \to \infty\})$
end if
For any $\lambda' \in \mathbb{D}$, let $c_\lambda[\lambda = \lambda']$ the capacity function $E \to \mathbb{R}$ of $G_{\lambda'}$
$S_{\lambda_{\min}}$ = MinimalMinCut($G, c_\lambda[\lambda = \lambda_{\min}]$)
$S_{\lambda_{\max}}$ = MinimalMinCut($G, c_\lambda[\lambda = \lambda_{\max}]$)
if $|S_{\lambda_{\min}}| > |V|/2$ **then**
  return ApxParametricMinCut(Contract($G, c_\lambda, S_{\lambda_{\min}}$), $\lambda_{\min}, \lambda_{\max}$)
end if
if $|S_{\lambda_{\max}}| < |V|/2$ **then**
  return ApxParametricMinCut(Contract($G, c_\lambda, V \setminus S_{\lambda_{\max}}$), $\lambda_{\min}, \lambda_{\max}$)
end if
$(\lambda_1, \lambda_2) := (\lambda_{\min}, \lambda_{\max})$
while $\lambda_2 - \lambda_1 > \varepsilon$ **do**
  $\lambda' := \lambda_1 + \lfloor (\lambda_2 - \lambda_1)/2\varepsilon \rfloor \cdot \varepsilon$
  $S_{\lambda'}$ = MinimalMinCut($G, c_\lambda[\lambda = \lambda']$)
  if $|S_{\lambda'}| \ge |V|/2$ **then**
    $\lambda_2 := \lambda'$
  else
    $\lambda_1 := \lambda'$
  end if
end while
For $i = 1, 2$, S_{λ_i} := MinimalMinCut($G, c_\lambda[\lambda = \lambda_i]$)
(Λ_1, τ_1) = ApxParametricMinCut(Contract($G, c_\lambda, V \setminus S_{\lambda_1}$), $\lambda_{\min}, \lambda_1$)
(Λ_2, τ_2) = ApxParametricMinCut(Contract($G, c_\lambda, S_{\lambda_2}$), $\lambda_2, \lambda_{\max}$)
$\Lambda :=$ **if** $|S_{\lambda_1}| = |S_{\lambda_2}|$ **then** $\Lambda_1 \cup \Lambda_2$ **else** $\Lambda_1 \cup \{\lambda_2\} \cup \Lambda_2$
$\tau := \{v \in S_{\lambda_1} \to \tau_1(v), v \in V \setminus S_{\lambda_2} \to \tau_2(v), v \in S_{\lambda_2} \setminus S_{\lambda_1} \to \lambda_2\}$
return (Λ, τ)

The main idea of the procedure ApxParametricMinCut is to find the (approximately) most balanced minimal s,t-cuts S_{λ_1} and S_{λ_2} and use them to reduce the problem size in the recursive calls significantly. Specifically, we want to find such $\lambda_1 \le \lambda_2$ that $|S_{\lambda_1}| \le n/2$, $|S_{\lambda_2}| \ge n/2$ and $\lambda_2 - \lambda_1 = \varepsilon$.

Suppose $n > 2$ as otherwise the problem is trivial. First, we compute minimal min-cuts in $G_{\lambda_{\min}}$ and $G_{\lambda_{\max}}$. This takes two max-flow runs, i.e., $T_{\text{maxflow}}(n, m)$ time, plus $O(m)$ time by Lemma C.1.

It might happen that $|S_{\lambda_{\min}}| \le |S_{\lambda_{\max}}| < n/2$ or $n/2 < |S_{\lambda_{\min}}| \le |S_{\lambda_{\max}}|$. In these special cases we can immediately reduce the vertex set by a factor of at least two by contracting $T_{\lambda_{\max}}$ or $S_{\lambda_{\min}}$ respectively, and recurse on the reduced graph. By Lemma C.7 and the definition of $\lambda_{\min}, \lambda_{\max}$ this reduction does not influence the structure of parametric cuts.

So suppose $|S_{\lambda_{\min}}| \le n/2$ and $|S_{\lambda_{\max}}| \ge n/2$. Set $\lambda_1 = \lambda_{\min}$ and $\lambda_2 = \lambda_{\max}$. So we have $|S_{\lambda_1}| \le n/2$ and $|S_{\lambda_2}| \ge n/2$ initially. We maintain this invariant and gradually shrink the interval $[\lambda_1, \lambda_2]$ until its length gets precisely ε in a binary search-like way. We repeatedly try the pivot $\lambda' = \lambda_1 + \lfloor (\lambda_2 - \lambda_1)/2\varepsilon \rfloor \cdot \varepsilon$ and compute $S_{\lambda'}$. If $|S_{\lambda'}| \ge n/2$, we set $\lambda_2 = \lambda'$, and otherwise we set $\lambda_1 = \lambda'$. Note that $\lambda_2 - \lambda_1$ remains an integer multiple of ε at all times. The whole process costs $O(\log[(\lambda_{\max} - \lambda_{\min})/\varepsilon]) = O(\log(R/\varepsilon))$ max-flow executions.

```
function MinimalMinCut(G = (V, E), c : E → ℝ)
f = MaxFlow(G, s, t, c)
return {v ∈ V : v reachable from s in the residual network G_f}

function Contract(G = (V, E), c_λ : E → 𝔻 → ℝ, X ⊆ V)                // |X ∩ {s, t}| = 1
w* := if s ∈ X then s else t
V' := V \ X ∪ {w*}
E' := ∅
c'_λ := E' → 𝔻 → ℝ
for uv ∈ E do
    u' := if u ∈ X then w* else u
    v' := if v ∈ X then w* else v
    if (u', v') ≠ (s, t) then
        if u'v' ∉ E' then
            E' := E' ∪ {u'v'}
            c'_λ(u'v') := c_λ(uv)
        else
            c'_λ(u'v') := c'_λ(u'v') + c_λ(uv)                    // We add functions here.
        end if
    end if
end for
return (G' = (V', E'), c'_λ)
```

Let $G_1 = G/T_{\lambda_1}$ and $G_2 = G/S_{\lambda_2}$. Note that G_1, G_2 may contain parallel edges or a direct st edge as a result of contraction. Hence, these graphs are first preprocessed by (1) removing self-loops and direct st edges, (2) merging parallel edges by summing their cost functions. The contraction and preprocessing is performed using the procedure Contract. Note that none of these preprocessing steps change the minimal cuts of $G_{1,\lambda}$ or $G_{2,\lambda}$ for any λ: the direct st edges cross *all* s,t-cuts.

Next, we recursively compute ε-approximate parametric min s,t-cut in graphs $G_1 = G/T_{\lambda_1}$ and $G_2 = G/S_{\lambda_2}$, The recursive call on G_1 is made with $(\lambda_{\min}, \lambda_{\max})$ set to $(\lambda_{\min}, \lambda_1)$, whereas the recursive call on G_2 uses $(\lambda_{\min}, \lambda_{\max}) := (\lambda_2, \lambda_{\max})$. Note that indeed we have $G_{1,\lambda_1} = G_{1,\lambda'}$ for $\lambda' > \lambda_1$ as required since the t-side of the minimal min s,t-cut in G_{1,λ_1} contains only t. Similarly, $G_{2,\lambda_2} = G_{2,\lambda'}$ for all $\lambda' < \lambda_2$.

Let $(\Lambda_1, \tau_1) = (\{\lambda_{1,1}, \ldots, \lambda_{1,a}\}, \tau_1)$ and $(\Lambda_2, \tau_2) = (\{\lambda_{2,1}, \ldots, \lambda_{2,b}\}, \tau_2)$ be the returned ε-approximate parametric min s,t-cuts of G_1 and G_2 respectively. We return (Λ, τ) as the ε-approximate parametric min-s,t-cut of G, where

$$\Lambda = \begin{cases} \Lambda_1 \cup \Lambda_2 & \text{if } |S_{\lambda_1}| = |S_{\lambda_2}| = n/2, \\ \Lambda_1 \cup \{\lambda_2\} \cup \Lambda_2 & \text{otherwise.} \end{cases} \qquad \tau(v) = \begin{cases} \tau_1(v) & \text{if } v \in S_{\lambda_1}, \\ \tau_2(v) & \text{if } v \in T_{\lambda_2}, \\ \lambda_2 & \text{otherwise.} \end{cases}$$

Let us now prove the correctness of this algorithm. We proceed by induction on n. For $n \leq 2$ this is trivial, so suppose $n > 3$ and that recursive calls are made. Clearly, $\lambda_{1,a} \leq \lambda_1 < \lambda_2 < \lambda_{2,1}$.

Item (3) of Definition C.5 follows easily by induction and the definition of λ_1, λ_2. Let $\Lambda = \{\lambda'_1, \ldots, \lambda'_k\}$. That $S(\lambda'_i) = \{v \in V : \tau(v) \leq \lambda'_i\}$ is a minimal min s,t-cut of $G_{\lambda'_i}$ for all $\lambda'_i \in \Lambda$ (i.e., item (2) of Definition C.5) follows directly by Lemma C.7 the definitions of λ_1, λ_2.

Now consider item (1) of Definition C.5. For some $j < k$ we have $\lambda'_j = \lambda_{1,a}$. For all $i = 0, \ldots, k-1$, $i \neq j$, item (1), i.e., that $S_{\lambda'_i}$ is a minimal min s,t-cut for all $\lambda' \in [\lambda'_i, \lambda'_{i+1})$, follows directly inductively.

If $|S_{\lambda_1}| = |S_{\lambda_2}| = n/2$, then $S_{\lambda_{1,a}} = S_{\lambda_2}$. By induction it follows that $S_{\lambda_{1,a}}$ is a minimal min s,t-cut of $G_{\lambda'}$ for all $\lambda' \in [\lambda_2, \lambda_{2,1} - \varepsilon)$, and thus also for all $\lambda' \in [\lambda_{1,a}, \lambda_{2,1} - \varepsilon) = [\lambda'_j, \lambda'_{j+1} - \varepsilon)$.

If, on the other hand, $S_{\lambda_1} \subsetneq S_{\lambda_2}$, then $\lambda'_{j+1} = \lambda_2$. Since $S_{\lambda_{1,a}} = S_{\lambda_1}$, $S_{\lambda_{1,a}}$ is indeed a minimal min s,t-cut for all $\lambda' \in [\lambda_{1,a}, \lambda_2 - \varepsilon) = [\lambda'_j, \lambda'_{j+1} - \varepsilon)$ as $\lambda_2 - \varepsilon = \lambda_1$.

Note that the input graph of each of the recursive calls has at most $n/2+1$ vertices. Moreover, by merging the parallel edges (and summing their costs) after the contraction we can guarantee that $|E(G_1)|+|E(G_2)| \leq |E(G)|+n/2$. Indeed, observe that the only edges of G_2 that can also appear in G_1 are those incident to s in G_2, and there are at most $|T_{\lambda_2}| \leq n/2$ of them.

There is one important technical detail here: even though the individual functions $c_\lambda(uv)$ (for the edges uv of the original input graph G) can be evaluated in constant time, after summing k of such functions in the process this cost can be as much as $\Theta(k)$. We now argue that this cannot happen in our case due to preprocessing G_1 and G_2, and the evaluation cost is $O(1)$ for all edges in all recursive calls. More concretely, one can show that each edge capacity function in a recursive call can be expressed as the sum of at most one original capacity function $c_\lambda(xy)$ and a real number. Indeed, suppose this is the case for some call with input G. Then, each edge uv of G_1 either (1) is contained in G and has not resulted from merging some parallel edges after contraction, (2) has $u \notin \{s,t\}$ and $v=t$ and resulted from merging edges $uz_1,\ldots,uz_l$ such that $z_1,\ldots,z_l \in T_{\lambda_1}$. The former case is trivial. In the latter case, for at least $l-1$ of these z_i we have $z_i \neq t$, so $c_\lambda(uz_i)$ is a constant function. Fot at most one z_j is of the form $c_\lambda(xy)+\Delta$ for some original capacity function $c_\lambda(xy)$ and $\Delta \in \mathbb{R}$. We conclude that the capacity function of uv in G_1 is of the same form and equals $c_\lambda(xy)+\Delta'$, where $\Delta' = \Delta + \sum_{i\neq j} c_\lambda(uz_i) \in \mathbb{R}$. The proof for G_2 is analogous.

Now let us analyze the running time of the algorithm. One can easily inductively prove that:

- Each graph at the i-th level of the recursion tree has less than $n/2^i+2$ vertices; hence, there are no more than $\log_2 n+1$ levels in the tree.
- The sum n_i of numbers of vertices through all the graphs at the i-th level is less than $2^i(n/2^i+2) \leq n+2^{i+1} \leq 3n$.
- Since, the sum m_i of numbers of edges in graphs at level $i>0$ satisfies $m_i \leq m_{i-1}+n_{i-1}/2 \leq m_{i-1}+3n/2$, we have $m_i \leq m+3in/2 = O(m+n\log n)$.

By the above, and since the function T_{maxflow} is convex, we conclude the total time cost at the i-th level is $O\left(T_{\text{maxflow}}(n, m+n\log n)\log(R/\varepsilon)\right)$. Recall that there are $O(\log n)$ levels and therefore the total time is $O\left(T_{\text{maxflow}}(n, m+n\log n)\log(R/\varepsilon)\log n\right)$.

C.1. Exact Parametric Min s,t-Cut

In this section we show how Theorem 3.2 implies new bounds on computing *exact* parametric min s,t-cuts in a few interesting settings.

Integer Polynomial Costs. Suppose all the parametric costs $c_\lambda(uv)$ are of the form $c_\lambda(uv) = Q_{uv}(\lambda)$, where each Q_{uv} is a (possibly different) constant-degree polynomial with integer coefficients bounded in the absolute value by an integer $U>0$ and take nonnegative values on $\mathbb{D}$. Recall Q_{uv} can have a positive degree only if $u=s$ or $v=t$. Moreover, if $u=s$, then Q_{uv} is increasing, whereas when $v=t$, then Q_{uv} is decreasing.

Observe that the parametric capacity $c_\lambda(S)$ of any S, $s \in S \subseteq V \setminus \{t\}$ is a constant-degree polynomial with integer coefficients bounded by nU in absolute value. The same applies to a difference polynomial $c_\lambda(S)-c_\lambda(S')$ for any two such sets S, S'.

It is known that for a constant-degree polynomials Q with integer coefficients bounded by W:

- The roots of Q are of absolute value $O(\text{poly}(W))$ (e.g., (Yap et al., 2000)).
- Any two distinct roots of Q are at least $\Omega(1/\text{poly}(W))$ apart. (Mahler, 1964)

This means that by setting $\lambda_{\min} = -R/2$ and $\lambda_{\max} = R/2$ (or slightly less aggressively, if e.g., $R/2 \notin \mathbb{D}$) for a sufficiently large even integer $R = O(\text{poly}\, nU)$ such that $R/2$ exceeds the maximum possible absolute value of a root of any polynomial of the form $c_\lambda(S)-c_\lambda(S')$, we will indeed have $G_{\lambda_{\min}} = G_{\lambda'}$ for all $\lambda' < \lambda_{\min}$ and $G_{\lambda_{\max}} = G_{\lambda'}$ for all $\lambda' > \lambda_{\max}$.

Moreover, assume we compute an ε-approximate parametric min s,t-cut (Λ, τ), where $\Lambda = \{\lambda_1,\ldots,\lambda_k\}$. Suppose for some i there exists λ', $\max(\lambda_i, \lambda_{i+1}-2\varepsilon) \leq \lambda' < \lambda_{i+1}$, such that the minimal

s, t-cut $S_{\lambda'}$ in $G_{\lambda'}$ satisfies $S_{\lambda_i} \subsetneq S_{\lambda'} \subsetneq S_{\lambda_{i+1}}$. Let $\lambda_i^* = \max(\lambda_i, \lambda_{i+1} - 2\varepsilon)$. Note that $S_{\lambda_i} = S_{\lambda_i^*}$ by Definition C.5. Since $S_{\lambda'}$ is minimal, $c_\lambda(S_{\lambda_i})(\lambda_i^*) - c_\lambda(S_{\lambda'})(\lambda_i^*) \leq 0$ and $c_\lambda(S_{\lambda_i})(\lambda') - c_\lambda(S_{\lambda'})(\lambda') > 0$. So the polynomial $c_\lambda(S_{\lambda_i}) - c_\lambda(S_{\lambda'})$ is non-zero and has a root in the interval $[\lambda_i^*, \lambda')$. Similarly one can prove that the polynomial $c_\lambda(S_{\lambda'}) - c_\lambda(S_{\lambda_{i+1}})$ is non-zero and has a root in the interval $[\lambda', \lambda_{i+1})$. We conclude that the product polynomial $[c_\lambda(S_{\lambda_i}) - c_\lambda(S_{\lambda'})] \cdot [c_\lambda(S_{\lambda'}) - c_\lambda(S_{\lambda_{i+1}})]$ is non-zero, has constant degree, has integer coefficients of order $O(\text{poly}\, nU)$, and has two distinct roots in the interval $[\lambda_i^*, \lambda_{i+1})$, i.e., less than 2ε apart. Therefore, if we set ε so that $1/\varepsilon$ is a sufficiently large integer but still polynomial in nU, the assumption $S_{\lambda_i} \subsetneq S_{\lambda'} \subsetneq S_{\lambda_{i+1}}$ leads to a contradiction. As a result, for all such λ', $S_{\lambda'}$ equals either $S_{\lambda'}$ or $S_{\lambda''}$.

In other words, computing an ε-approximate min s, t-cut (Λ, τ), where $\Lambda = \{\lambda_1, \ldots, \lambda_k\}$, gives as the structure of all possible minimal min s, t-cuts S_λ in the following sense. Suppose $\Lambda^* = \{\lambda_1^*, \ldots, \lambda_l^*\}$ is an *exact* parametric min s, t-cut. Then $k = l$ and $S(\lambda_i) = \{v \in V : \tau(v) \leq \lambda_i\} = S_{\lambda_i^*}$ for all $i = 1, \ldots, k$. To compute λ_i^*, it is hence enough to find the *unique* $\lambda_i^* \in (\lambda_{i-1}, \lambda_i]$ such that $c_\lambda(S(\lambda_{i-1}))(\lambda_i^*) = c_\lambda(S(\lambda_i))(\lambda_i^*)$ which boils down to solving a polynomial equation of constant degree. It is well known that such equations can be solved exactly in constant time for degrees at most 4.

Observe that if we run our ε-approximate parametric min s, t-cut algorithm with $\lambda_{\min}, \lambda_{\max}, \varepsilon$ set as described above, maximum flow is always invoked on some minor H of G_λ for λ that is an integer multiple of ε. Since, $1/\varepsilon$ is an integer, by multiplying edge costs in H by $1/\varepsilon$, we only need a maximum flow algorithm that can handle integer edge capacities of order $O(\text{poly}(nU))$. By plugging in the best-known algorithms for computing max flow with integral capacities, we obtain the following.

Theorem C.9. *Let G be a graph whose parameterized capacities are constant-degree polynomials with integer coefficients in $[-U, U]$. The structure of parametric min s, t-cut on G can be computed in:*

- $O(m \cdot \min(m^{1/2}, n^{2/3}) \cdot \text{polylog}\{n, U\})$ *time using a combinatorial algorithm (Goldberg & Rao, 1998),*
- $O((m + n^{1.5}) \cdot \text{polylog}\{n, U\})$ *time using the algorithm of (van den Brand et al., 2021),*
- $O(m^{1.497}\, \text{polylog}\{n, U\})$ *time using the algorithm of (Gao et al., 2021).*

The cut function can be found exacly in additional $O(n)$ time if the degrees of capacity polynomials are at most 4.

Discrete Domains. Let us now consider the case when $\mathbb{D}$ is discrete and has ℓ elements. Suppose all parametric costs are arbitrary functions meeting the requirements of the parametric min s, t-cut problem. Then, we can make the ε-approximate algorithm exact by employing the following simple modifications. We start with $\lambda_{\min} = \min \mathbb{D}$ and $\lambda_{\max} = \max \mathbb{D}$. In the binary-search like step, we always choose the middle element of $\mathbb{D} \cap [\lambda_1, \lambda_2]$ as the next pivot. This way, all the max-flow computations are performed on minors of G_λ for $\lambda \in \mathbb{D}$.

Theorem C.10. *Let G be a graph with arbitrary parameterized capacities $\mathbb{D} \to \mathbb{R}$ for a discrete domain $\mathbb{D} \subseteq \mathbb{R}$, where $\ell = |\mathbb{D}|$. Let $T_{\text{maxflow}}(n', m')$ be defined as in Theorem 3.2. Then exact parametric min s, t-cut on G can be computed in $O(T_{\text{maxflow}}(n, m \log n) \cdot \log \ell \cdot \log n)$ time.*

Planar Graphs. By Remark C.8, all the max-flow computations in the algorithm of Theorem 3.2 are performed on minors of G. As a result, if the input graph G is planar, we can use state-of-the-art planar max s, t-flow algorithms to obtain better bounds on the parametric min-s, t-cut algorithms on planar graphs. Since maximum s, t-flow for planar graphs can be computed in $O(n \log n)$ time even for real capacities (Borradaile & Klein, 2009; Erickson, 2010), planar parametric min s, t-cut can be solved exactly:

- in $O(n\, \text{polylog}\{n, U\})$ time when parameterized capacities are constant degree polynomials with integer coefficients in $[-U, U]$,
- in $O(n \log^3 n \log \ell)$ time for discrete domains $\mathbb{D} \subseteq \mathbb{R}$ of size ℓ.

What may be surprising, our reduction is powerful enough to obtain an interesting subquadratic *strongly polynomial* exact algorithm computing parametric min s, t-cut in a planar graph with capacity functions that are arbitrary polynomials of degree no more than 4 and real coefficients.

We now sketch this algorithm. It is based on the *parametric search* technique (Megiddo, 1983) (see also (Agarwal et al., 1994)). Suppose we want to solve some decision problem $\mathcal{P}(\alpha)$, where $\alpha \in \mathbb{R}$, such that if $\mathcal{P}(\alpha_0)$ is a yes instance, then $\mathcal{P}(\alpha')$ for all $\alpha'' < \alpha_0$ is also a yes instance. We wish to find the maximum α^* for which $\mathcal{P}(\alpha^*)$ is a yes instance. An example problem $\mathcal{P}(\alpha)$ could be *"does an s,t-flow of value α exist in G?"*. Then, α^* clearly equals the maximum flow in G.

Suppose we have an efficient strongly polynomial algorithm solving the decision problem. Then, in practice one could find α^* via binary search given some initial interval containing α^*; however, in general this would not lead to an exact algorithm for real values of α^*. Parametric search is a technique for converting a strongly polynomial *parallel* decision algorithm into a *sequential or parallel* strongly polynomial *optimization* algorithm as explained above. The only requirement to keep in mind is that the decision algorithm is governed by comparisons, each of which amounts to testing the sign of some low-degree (say, no more than 4) polynomial in α. Specifically, suppose we have a parallel decision algorithm $\mathcal{A}$ that uses $W_{\mathcal{A}}$ work and $D_{\mathcal{A}}$ depth, and also a (possibly the same) another parallel decision algorithm $\mathcal{B}$ with $W_{\mathcal{B}}$ work and $D_{\mathcal{B}}$ depth. Suppose for simplicity all these quantities are polynomial in n. Then, parametric search yields a strongly-polynomial optimization algorithm computing α^* in $\tilde{O}(D_{\mathcal{A}} \cdot W_{\mathcal{B}} + W_{\mathcal{A}})$ work and $\tilde{O}(D_{\mathcal{A}} \cdot D_{\mathcal{B}})$ depth.

Now back to planar graphs. We will use parametric search in a nested way. First of all, we will need a decent parallel max flow algorithm for planar graphs. It is well-known (e.g., (Erickson, 2010)) that the decision variant of the max s,t-flow problem on planar graphs is reducible to negative cycle detection in the dual graph (which is also planar). There exists a parallel negative cycle detection algorithm on planar graphs with $\tilde{O}(n + n^{3/2}/d^3)$ work and $\tilde{O}(d)$ depth for any $d \geq 1$ (Karczmarz & Sankowski). Hence, by using that algorithm as both $\mathcal{A}$ (for $d = D^{3/7}$) and $\mathcal{B}$ (for $d = D^{4/7}$), where D is a parameter, we have $W_{\mathcal{A}} = \tilde{O}(n + n^{3/2}/D^{9/7})$, $D_{\mathcal{A}} = \tilde{O}(D^{3/7})$, $W_{\mathcal{B}} = \tilde{O}(n + n^{3/2}/D^{12/7})$, $D_{\mathcal{B}} = \tilde{O}(D^{4/7})$. So parametric search yields a strongly-polynomial parallel max-flow algorithm for planar graphs with work $\tilde{O}(n + n^{3/2}/D^{9/7})$ and depth $\tilde{O}(D)$ for any $D \geq 1$.

Given a parallel algorithm for max flow in planar graphs, we can use parametric search (instead of binary search) once again when computing the pair λ_1, λ_2 in our recursive algorithm. More specifically, we would like λ_1 to be the largest such that $|S_{\lambda_1}| \leq n/2$, whereas λ_2 to be the smallest such that $|S_{\lambda_2}| \geq n/2$. It is easy to see that λ_1 and λ_2 are precisely neighboring (or the same) breakpoints of the cut function, i.e., belong to Λ from Definition C.4. To actually compute λ_1, λ_2, we use parametric search with $\mathcal{A}$ set to the obtained parallel max-flow algorithm[5], and $\mathcal{B}$ to the best known algorithm that computes a minimum min s,t-cut in a planar graph, i.e., a combination of the max-flow algorithm of (Borradaile & Klein, 2009; Erickson, 2010), and linear time graph search. So, in the outer parametric search instance we have $W_{\mathcal{A}} = \tilde{O}(n + n^{3/2}/D^{9/7})$, $D_{\mathcal{A}} = \tilde{O}(D)$, and $W_{\mathcal{B}} = \tilde{O}(n)$. Therefore, the obtained algorithm runs in $\tilde{O}(n^{3/2}/D^{9/7} + Dn)$ sequential time. By setting $D = n^{7/32}$, we obtain $\tilde{O}(n^{1+7/32}) = \tilde{O}(n^{1.21875})$ time.

We stress that all the algorithms (Borradaile & Klein, 2009; Erickson, 2010; Karczmarz & Sankowski) used above proceed by only adding and comparing edge weights. Adding polynomials cannot increase their degrees, so indeed when these algorithms are run "generically" for some λ, the control flow depends only on signs of some small degree polynomials.

Theorem C.11. *Let G be a planar graph whose parameterized capacities are all polynomials of degree at most 4 with real coefficients. There exists a strongly polynomial algorithm computing parametric min s,t-cut in G exactly in $\tilde{O}(n^{1+7/32})$ time.*

D. Removing Assumptions on F_i

In this section we argue why the assumptions on the functions F_i introduced in Section 2 are valid without loss of generality. More precisely, we assumed that for all i, $F_i(\emptyset) = F_i(V_i) = 0$ and $F_i(S) \geq 0$ for all S. Here we show that a simple preprocessing step can enforce all of these conditions.

Without changing the original problem we can shift each F_i such that it evaluates to 0 on $\emptyset$, by defining $\overline{F}_i(S) = F_i(S) - F_i(\emptyset)$. This only changes F by a constant term without affecting the sets that minimize the parametric problem (1).

For each i, we use Lemma D.1 to find a point $w_i \in B(\overline{F}_i)$. Using this point, we define $\overline{\overline{F}}_i(S) = \overline{F}_i(S) - w_i(S)$. Since by definition $w_i(V_i) = \overline{F}_i(V_i)$, we have that $\overline{\overline{F}}_i(V_i) = 0$. Also, we have $\overline{\overline{F}}_i(\emptyset) = \overline{F}_i(\emptyset) = 0$. Finally, since $w_i(S) \leq \overline{F}_i(S)$ for all S, we also have $\overline{\overline{F}}_i(S) \geq 0$.

[5] Actually, it is computing max-flow followed by a graph search to determine the minimal min s,t-cut. However, this latter step does not involve any comparisons on capacities, so its depth can be ignored.

Now we can equivalently rewrite the parametric problem

$$\begin{aligned}F_\alpha(A) &= F(A) + \sum_{j\in A} \psi'_j(\alpha) \\ &= \overline{F}(A) + \sum_{j\in A} \psi'_j(\alpha) + \left(\sum_{i=1}^{m} F_i(\emptyset)\right) \\ &= \overline{\overline{F}}(A) + \sum_{j\in A} \left(\psi'_j(\alpha) + \sum_{i=1}^{r} w_i(j)\right) + \left(\sum_{i=1}^{m} F_i(\emptyset)\right).\end{aligned}$$

Now we can solve the problem on $\overline{\overline{F}} = \sum_{i=1}^r \overline{\overline{F}}_i$ with the parametric penalties $\overline{\overline{\psi}}'_j(\alpha) = \psi'_j(\alpha) + \sum_{i=1}^r w_i(j)$, which maintain the validity of Assumption 2.6.

To compute a point in the base polytope of a submodular function we use the following folklore lemma, which shows that the running time of our initialization procedure is $O(\sum_{i=1}^r |V_i| \cdot \mathrm{EO}_i)$:

Lemma D.1 ((Fujishige, 1980))**.** *Let $F : 2^V \to \mathbb{Z}$ be a submodular set function, with $F(\emptyset) = 0$, and let $B(F)$ be its base polytope. Given any $x \in \mathbb{R}^{|V|}$, one can compute*

$$\arg\max_{w\in B(F)} \langle x, w\rangle$$

using $O(|V|)$ calls to an evaluation oracle for F. Furthermore w is integral.

E. Deferred Proofs

E.1. Proof of Lemma 4.3

We define the primal and dual optima of this problem, which will be useful for the proof.

Definition E.1 (Graph subproblem minimizers)**.** *Let $\widetilde{x}^*$ be the minimizer of*

$$\min_x g(x) + \phi(x)\,, \tag{8}$$

and $\widetilde{w}^*$ be the minimizer of

$$\min_{w\in B(G)} \phi^*(-w)\,. \tag{9}$$

The main tool that we will use for this proof will be the following two structural statements, which can be extracted from Propositions 4.2 and 8.3 in (Bach, 2011).

Lemma E.2 ((Bach, 2011))**.** *Consider any submodular function $F : 2^V \to \mathbb{R}$.*

1. *Fix some $x \in \mathbb{R}^n$. For any $w \in \mathbb{R}^n$, w is an optimizer of $\max_{w\in B(F)} \langle w, x\rangle$ if and only if there exists a permutation π of $[n]$ such that $x_{\pi_1} \geq x_{\pi_2} \geq \cdots \geq x_{\pi_n}$ and for all $u \in V$ we have*

$$w_u = \begin{cases} F(\{\pi_1\}) & \text{if } u = 1\,, \\ F(\{\pi_1, \pi_2, \dots, \pi_u\}) - F(\{\pi_1, \pi_2, \dots, \pi_{u-1}\}) & \text{if } u \geq 2\,. \end{cases}$$

2. *Given a function ϕ that satisfies the conditions in Definition 2.6, the optimal solution $\widetilde{x}^*$ to the problem*

$$\min_x f(x) + \phi(x)\,,$$

where f is the Lovász extension of F, is given by

$$\widetilde{x}^*_u = -\inf(\{\lambda \in \mathbb{R} \,:\, u \in S(\lambda)\})$$

for all $u \in V$, where

$$S(\lambda) = \underset{S \subseteq V}{\operatorname{argmin}}\, F(S) + \sum_{u \in S} \phi'_u(-\lambda)\,.$$

Additionally, we present two simple lemmas which will be useful in the proof. The first one upper bounds the ℓ_1 diameter of a base polytope, and the second one upper bounds the ℓ_1 norm of the gradient of a function in the base polytope.

Lemma E.3. *For any submodular function $F : 2^V \to \mathbb{R}_{\geq 0}$ and $F(S) \leq F_{\max}$ for all $S \subseteq V$, we have that*

$$\max_{w \in B(F)} \|w\|_1 \leq 2nF_{\max}\,.$$

Proof. By definition of $B(F)$, for all $u \in V$ we have $w_u \leq F(\{u\}) \leq F_{\max}$, so $\sum\limits_{u \in V : w_u \geq 0} w_u \leq nF_{\max}$. Also, $\sum\limits_{u \in V} w_u = F(V)$, so we conclude that

$$\begin{aligned}
\|w\|_1 &= \sum_{u \in V : w_u \geq 0} w_u - \sum_{u \in V : w_u < 0} w_u \\
&= 2 \sum_{u \in V : w_u \geq 0} w_u - F(V) \\
&\leq 2nF_{\max} - F(V) \\
&\leq 2nF_{\max}\,.
\end{aligned}$$

□

Lemma E.4. *For any submodular function $F : 2^V \to \mathbb{R}_{\geq 0}$, $F(S) \leq F_{\max}$ for all $S \subseteq V$, and function $\psi : \mathbb{R}^n \to \mathbb{R}$ satisfying the conditions of Definition 2.6 we have that*

$$\max_{w \in B(F)} \|\nabla \psi^*(-w)\|_1 \leq \frac{2nF_{\max}}{\sigma} + \|\nabla \psi^*(0)\|_1\,.$$

Proof.

$$\begin{aligned}
\|\nabla \psi^*(-w)\|_1 &\leq \|\nabla \psi^*(-w) - \nabla \psi^*(0)\|_1 + \|\nabla \psi^*(0)\|_1 \\
&\leq \frac{1}{\sigma}\|w\|_1 + \|\nabla \psi^*(0)\|_1 \\
&\leq \frac{2nF_{\max}}{\sigma} + \|\nabla \psi^*(0)\|_1\,,
\end{aligned}$$

where we used the triangle inequality, the $\frac{1}{\sigma}$-smoothness of the ψ_u^*'s, and Lemma E.3. □

We are now ready to proceed with the proof.

Proof of Lemma 4.3. We let $\Lambda = \{\lambda_1, \ldots, \lambda_k\}$, where $\lambda_1 < \cdots < \lambda_k$, and define $S(\lambda)$ to be a minimal set in

$$\underset{S \subseteq V}{\operatorname{argmin}}\, G(S) + \sum_{u \in S} \phi'_u(-\lambda)$$

for all $\lambda \in \mathbb{R}$. Note that this can be equivalently written as

$$\begin{aligned}
&\underset{S\subseteq V}{\operatorname{argmin}}\, G(S) + \sum_{u\in S} \max\{0, \phi_u'(-\lambda)\} - \sum_{u\in S} \max\{0, -\phi_u'(-\lambda)\} \\
&= \underset{S\subseteq V}{\operatorname{argmin}}\, G(S) + \sum_{u\in S} \max\{0, \phi_u'(-\lambda)\} + \sum_{u\in V\setminus S} \max\{0, -\phi_u'(-\lambda)\} - \sum_{u\in V} \max\{0, -\phi_u'(-\lambda)\} \\
&= \underset{S\subseteq V}{\operatorname{argmin}}\, G(S) + \sum_{u\in S} \max\{0, \phi_u'(-\lambda)\} + \sum_{u\in V\setminus S} \max\{0, -\phi_u'(-\lambda)\} \\
&= \underset{S\subseteq V}{\operatorname{argmin}}\, c_\lambda^+(S\cup\{s\})\,,
\end{aligned}$$

by the definition of the parametric capacities c_λ, where $c_\lambda^+(S\cup\{s\}) = \sum_{\substack{u\in S\cup\{s\}\\ v\in V\setminus(S\cup\{s\})}} c_\lambda(u,v)$. Additionally, we denote $\varepsilon = \frac{1}{3L}$ for convenience. By the second item of Lemma E.2, we know that the minimizer of $\min_x g(x) + \phi(x)$, where g is the Lovász extension of G, is defined as

$$\widetilde{x}_u^* = -\inf\{\lambda \in \mathbb{R} \,:\, u \in S(\lambda)\}\,.$$

For all $u \in V$, let $i_u = \underset{i\in[k]}{\operatorname{argmin}}\,\{\lambda_i \mid u \in S(\lambda_i)\}$ and $\widetilde{x}_u = -\lambda_{i_u}$. We will first prove that $\|\widetilde{x} - \widetilde{x}^*\|_\infty \le \varepsilon$. Now, by definition we have that $\widetilde{x}_u^* \ge \widetilde{x}_u$. Additionally, setting $\lambda_0 = -\infty$ for convenience, we have $u \notin S(\lambda_{i_u-1})$, and $u \in S(-\widetilde{x}_u^*)$, so $S(\lambda_{i_u-1}) \subset S(-\widetilde{x}_u^*)$. By the first item of Definition C.5, this implies that

$$-\widetilde{x}_u^* \ge \lambda_{i_u} - \varepsilon = -\widetilde{x}_u - \varepsilon \Leftrightarrow \widetilde{x}_u^* \le \widetilde{x}_u + \varepsilon\,.$$

Therefore, we have concluded that $|\widetilde{x}_u - \widetilde{x}_u^*| \le \varepsilon$ for all $u \in V$, i.e. $\|\widetilde{x} - \widetilde{x}^*\|_\infty \le \varepsilon$.

We compute a dual solution $\widehat{w} = -\nabla\phi(\widetilde{x})$. We will show that $\widetilde{w}^*$ can be retrieved by rounding $\widehat{w}$. Using the fact that ϕ_u's are L-smooth and the optimality condition $\widetilde{w}^* = -\nabla\phi(\widetilde{x}^*)$ from Lemma B.5, we get that

$$\|\widehat{w} - \widetilde{w}^*\|_\infty = \|\nabla\phi(\widetilde{x}) - \nabla\phi(\widetilde{x}^*)\|_\infty \le L\|\widetilde{x} - \widetilde{x}^*\|_\infty \le L\varepsilon = 1/3\,.$$

On the other hand, by optimality of $\widetilde{w}^*$, it is a maximizer of $\max_{w\in B(G)} \langle w, \widetilde{x}^*\rangle$. By the first item of Lemma E.2, there exists a permutation $\pi_1, \dots, \pi_n$ of V such that $\widetilde{w}_u^* = G(\{\pi_1, \dots, \pi_u\}) - G(\{\pi_1, \dots, \pi_{u-1}\})$ for $u \in V$. As G takes integral values, we have $\widetilde{w}_u^* \in \mathbb{Z}$ for all $u \in V$, and since $|\widehat{w}_u - \widetilde{w}_u^*| \le 1/3 < 1/2$, we can exactly recover $\widetilde{w}^*$ by rounding each entry of $\widehat{w}$ to the closest integer.

Our next goal is to compute a G-decomposition of $\widetilde{w}^*$, which we will do by computing an exact primal solution and then again applying the first item of Lemma E.2. Given $\widetilde{w}^*$, we can easily recover the primal optimum $\widetilde{x}^* = \nabla\phi(-\widetilde{w}^*)$. In order to recover a decomposition $\widetilde{w}^* = \sum_{i=1}^{r} \widetilde{w}^{*i}$, we use the well-known fact (Edmonds, 1970) that

$$\max_{w\in B(G)} \langle w, x\rangle = \max_{w^i\in B(G_i)} \sum_{i=1}^{r} \langle w^i, x\rangle\,,$$

so for any $i \in [r]$, $\widetilde{w}^{*i}$ necessarily maximizes

$$\max_{w^i\in B(G_i)} \langle w^i, \widetilde{x}^*\rangle\,.$$

Therefore, by the first item of Lemma E.2, $\widetilde{w}^{*i}$ can be recovered by sorting the entries of $\widetilde{x}^*$ in decreasing order, such that $\widetilde{x}_{\pi_1}^* \ge \widetilde{x}_{\pi_2}^* \ge \cdots \ge \widetilde{x}_{\pi_n}^*$ for some permutation π of V, and then setting

$$\widetilde{w}_u^{*i} = G_i(\{\pi_1, \dots, \pi_u\}) - G_i(\{\pi_1, \dots, \pi_{u-1}\}) \tag{10}$$

for all $u \in V$. Note that $\widetilde{w}^{*i}$'s are in $\mathbb{Z}^n$.

The runtime is dominated by the computation of the decomposition in (10), which involves computing prefix cuts for each G_i and by Lemma E.5 takes time $O\left(\sum_{i=1}^{r} |V_i|^2\right)$. Therefore, the total runtime is $O\left(n + \sum_{i=1}^{r} |V_i|^2\right)$. □

Lemma E.5 (Computing all prefix cut values)**.** *Given a graph $G(V, E, c)$ with $V = \{1, 2, \dots, n\}$, we can compute the values $c^+([u])$ for all $u \in [n]$ in time $O(n^2)$.*

Proof. We note that $c^+(\emptyset) = 0$ and for any $u \geq 1$ we have

$$c^+([u]) = c^+([u-1]) + \sum_{v=u+1}^{n} c_{uv} - \sum_{v=1}^{u-1} c_{vu}\,. \tag{11}$$

Therefore $c^+([u])$ can be computed in $O(n)$ given $c^+([u-1])$. As we apply (11) n times, the total runtime is $O(n^2)$. □

E.2. Proof of Lemma 4.4

We first prove the following lemma, which helps us bound the range of parameters for parametric min s, t-cut.

Lemma E.6. *Consider a graph $G(V \cup \{s,t\}, E, c \geq 0)$ and a function $\phi(x) = \sum_{u \in V} \phi_u(x_u)$ that satisfies Assumption 2.6. Additionally, let $G(S) = c^+(S \cup \{s\})$ for all $S \subseteq V$ be the cut function associated with the graph. For any $\lambda \in \mathbb{R}$, we set $S(\lambda)$ to be the smallest set that minimizes*

$$\min_{S \subseteq V} G(S) + \sum_{u \in S} \phi'_u(-\lambda)\,.$$

Let $\rho = \max_{u \in V} |\phi'_u(0)|$ and $G_{\max} = \max_{S \subseteq V} G(S)$. Then, $S(\lambda_{\min}) = \emptyset$ and $S(\lambda_{\max}) = V$, where $\lambda_{\min} = -2\frac{\rho + G_{\max}}{\sigma}$ and $\lambda_{\max} = 2\frac{\rho + G_{\max}}{\sigma}$.

Proof. We first note that by the σ-strong convexity of the ϕ_u's, and since $-\lambda_{\min} > 0 > -\lambda_{\max}$, we have that

$$\phi'_u(-\lambda_{\min}) \geq \phi'_u(0) + \sigma|\lambda_{\min}|\,.$$

and

$$\phi'_u(-\lambda_{\max}) \leq \phi'_u(0) - \sigma|\lambda_{\max}|\,.$$

Therefore for any $\emptyset \neq S \subseteq V$ we have

$$\begin{aligned}
G(S) + \sum_{u \in S} \phi'_u(-\lambda_{\min}) &\geq G(S) + \sum_{u \in S} (\phi'_u(0) + \sigma|\lambda_{\min}|) \\
&\geq G(S) - |S|\rho + |S|\sigma|\lambda_{\min}| \\
&= G(S) - |S|\rho + 2|S|\sigma\frac{\rho + G_{\max}}{\sigma} \\
&> G(S) + G_{\max} \\
&\geq G(\emptyset)\,,
\end{aligned}$$

where we used the fact that $G(S) \geq 0$ and $G(\emptyset) \leq G_{\max}$, so $S(\lambda_{\min}) = \emptyset$. Similarly, for any $S \subset V$ we have

$$\begin{aligned}
G(S) + \sum_{u \in S} \phi'_u(-\lambda_{\max}) &= G(S) + \sum_{u \in V} \phi'_u(-\lambda_{\max}) - \sum_{u \in V \setminus S} \phi'_u(-\lambda_{\max}) \\
&\geq G(S) + \sum_{u \in V} \phi'_u(-\lambda_{\max}) - \sum_{u \in V \setminus S} (\phi'_u(0) - \sigma|\lambda_{\max}|) \\
&\geq G(S) + \sum_{u \in V} \phi'_u(-\lambda_{\max}) + |V \setminus S| \left(\sigma|\lambda_{\max}| - \rho\right) \\
&= G(S) + \sum_{u \in V} \phi'_u(-\lambda_{\max}) + |V \setminus S| \left(2\sigma\frac{\rho + G_{\max}}{\sigma} - \rho\right) \\
&> G(S) + \sum_{u \in V} \phi'_u(-\lambda_{\max}) + G_{\max} \\
&\geq G(V) + \sum_{u \in V} \phi'_u(-\lambda_{\max})\,,
\end{aligned}$$

where we used the fact that $G(S) \geq 0$ and $G(V) \leq G_{\max}$, so $S(\lambda_{\max}) = V$. □

We are now ready for the proof.

Proof of Lemma 4.4. We first shift the polytope $B(F)$ so that w is translated to 0. Specifically, for all $S \subseteq V$, we let $\widehat{F}(S) = F(S) - w(S)$ and $\widehat{F}_i(S) = F_i(S) - w^i(S)$ for all $i \in [r]$. As we are just subtracting a linear function, $\widehat{F}$ and the $\widehat{F}_i$'s are still submodular functions, and $B(\widehat{F}) = B(F) - w$, $B(\widehat{F}_i) = B(F_i) - w^i$ for all $i \in [r]$. Note that $w^i \in B(F_i)$ implies that the $\widehat{F}_i$'s (and thus also $\widehat{F}$) are non-negative, since

$$\widehat{F}_i(S) = F_i(S) - w(S) \geq 0\,,$$

and additionally $\widehat{F}_i(\emptyset) = F_i(\emptyset) = 0$ and $\widehat{F}_i(V_i) = F_i(V_i) - w^i(V_i) = 0$ for all $i \in [r]$ (also implying $\widehat{F}(\emptyset) = \widehat{F}(V) = 0$).

We run the algorithm from Lemma A.5 on the $\widehat{F}_i$'s to obtain directed graphs $G_i(V, E, c^i \geq 0)$ whose (V_i-restricted) cut functions $G_i(S) = c^{i+}(S)$ α-approximate $\widehat{F}_i(S)$, where $\alpha = \max_{i \in [r]} \{|V_i|^2/4\}$. More specifically,

$$\frac{1}{\alpha}\widehat{F}_i(S) \leq G_i(S) \leq \widehat{F}_i(S) \text{ for all } S \subseteq V_i,\; G_i(V_i) = \widehat{F}_i(V_i) \tag{12}$$

and

$$\frac{1}{\alpha}\left(B(F_i) - w^i\right) = \frac{1}{\alpha}B(\widehat{F}_i) \subseteq B(G_i) \subseteq B(\widehat{F}_i) = B(F_i) - w^i\,, \tag{13}$$

We also define the graph $G(V, E, c \geq 0)$, where $c = \sum_{i=1}^{r} c^i$ and has cut function $G(S) = \sum_{i=1}^{r} G_i(S)$ for all $S \subseteq V$. Then,

$$\frac{1}{\alpha}\widehat{F}(S) \leq G(S) \leq \widehat{F}(S) \text{ for all } S \subseteq V,\; G(V) = \widehat{F}(V) \tag{14}$$

and

$$\frac{1}{\alpha}\left(B(F) - w\right) \subseteq B(G) \subseteq B(F) - w\,. \tag{15}$$

We absorb the linear term that we subtracted from F into the parametric function. Concretely, we define ϕ as $\phi(x) = \psi(x) + \langle w, x\rangle$ for all $x \in \mathbb{R}^n$. It is easy to see that $\phi(x)$ is coordinate-wise separable, as $\phi(x) = \sum_{u \in V} \phi_u(x_u)$ where $\phi_u(x_u) = \psi_u(x_u) + w_u x_u$ for all $u \in V$ and that it satisfies Assumption 2.6, since $\phi_u''(x_u) = \psi_u''(x_u)$ and

$$|\phi_u'(0)| = |\psi_u'(0) + w_u| \leq |\psi_u'(0)| + F(\{u\}) \leq |\psi_u'(0)| + F_{\max} = (n+r)^{O(1)}\,.$$

Additionally, the Fenchel dual of ϕ_u is a shifted version of ψ_u, i.e.

$$\phi_u^*(z) = \max_{y \in \mathbb{R}} zy - \phi(y) = \max_{w \in \mathbb{R}} zy - \psi(y) - w_u y = \max_{w \in \mathbb{R}} (z - w_u)y - \psi(y) = \psi_u^*(z - w_u)\,.$$

We will now run the algorithm from Lemma 4.3 on graphs G_i and parametric function ϕ to obtain a dual solution vector $\widetilde{w}$. We note that, as F_i's and w^i's take integer values, G_i's take integer values too. This algorithm takes a $\frac{1}{3L}$-approximate parametric min s,t-cut as input, which we first compute using Theorem 3.2, with range of parameters $[\lambda_{\min}, \lambda_{\max}]$ given by Lemma E.6. We have

$$\begin{aligned}
\lambda_{\max} - \lambda_{\min} &= 4\frac{\max_{u \in V}|\phi_u'(0)| + \max_{S \subseteq V} G(S)}{\sigma} \\
&\leq 4\frac{\max_{u \in V}|\phi_u'(0)| + \widehat{F}_{\max}}{\sigma} \\
&\leq 4\frac{\max_{u \in V}|\phi_u'(0)| + F_{\max} + \|w\|_1}{\sigma} \\
&\leq 4\frac{\max_{u \in V}|\phi_u'(0)| + (2n+1)F_{\max}}{\sigma} \\
&= (n+r)^{O(1)}\,,
\end{aligned}$$

where we used Lemma E.3 and the fact that the quantities $|\phi'_u(0)|, F_{\max}, \frac{1}{\sigma}$ are bounded by $(n+r)^{O(1)}$ (Assumption 2.6). Based on Theorem 3.2, the time to obtain the $\frac{1}{3L}$-approximate parametric min s,t-cut will be

$$O\left(T_{\text{maxflow}}(n, |E'|\log n)\log\frac{\lambda_{\max}-\lambda_{\min}}{1/3L}\log n\right) = \widetilde{O}\left(T_{\text{maxflow}}\left(n, n+\sum_{i=1}^{r}|V_i|^2\right)\right).$$

So, by applying Lemma 4.3, we obtained a dual solution $\widetilde{w} = \sum_{i=1}^{r}\widetilde{w}^i$ for which $\widetilde{w}^i \in B(G_i)$ and

$$\widetilde{w} = \underset{\widetilde{w}\in B(G)}{\operatorname{argmin}}\,\phi^*(-\widetilde{w}) = \underset{\widetilde{w}\in B(G)}{\operatorname{argmin}}\,\psi^*(-w-\widetilde{w}). \tag{16}$$

For all $u \in V$ and $i \in [r]$, we set $w'^i_u = w^i_u + \widetilde{w}^i_u$ and $w'_u = \sum_{i=1}^{r} w'^i_u$. Note that these quantities are still integral. We now prove the two parts of the lemma statement (feasibility and optimality) separately.

Feasibility. For any $i \in [r]$ and $S \subseteq V_i$, we have that

$$w'^i(S) = w^i(S) + \widetilde{w}^i(S) \le w^i(S) + G_i(S) \le w^i(S) + \widehat{F}_i(S) = w^i(S) + F_i(S) - w^i(S) = F_i(S),$$

where the first inequality follows from the fact that $\widetilde{w}^i \in B(G_i)$. Similarly, we have

$$w'^i(V_i) = w^i(V_i) + \widetilde{w}^i(V_i) = w^i(V_i) + G_i(V_i) = w^i(V_i) + \widehat{F}_i(V_i) = F_i(V_i).$$

So we conclude that $w'^i \in B(F_i)$ for all $i \in [r]$.

Optimality. Let's set $h(z) := \psi^*(-z)$ for all $z \in \mathbb{R}^n$ for notational convenience, so that our goal is to prove that

$$h(w') - h(w^*) \le \left(1 - \frac{1}{\alpha}\right)(h(w) - h(w^*)).$$

We will prove a slightly different statement where w' is replaced by a solution on the path from w to w^*, which is enough because w' is optimal in $w + B(G)$. Concretely, we set $\bar{w} = \frac{1}{\alpha}(w^* - w)$ and instead will prove

$$h(w+\bar{w}) - h(w^*) \le \left(1 - \frac{1}{\alpha}\right)(h(w) - h(w^*)).$$

Now, h is a convex function, so applying convexity twice we have

$$\begin{aligned} h(w) &\ge h(w+\bar{w}) + \langle \nabla h(w+\bar{w}), -\bar{w}\rangle \\ &= h(w+\bar{w}) - \frac{1}{\alpha}\langle \nabla h(w+\bar{w}), w^* - w\rangle \end{aligned} \tag{17}$$

and

$$\begin{aligned} h(w^*) &\ge h(w+\bar{w}) + \langle \nabla h(w+\bar{w}), w^* - w - \bar{w}\rangle \\ &= h(w+\bar{w}) + \frac{\alpha-1}{\alpha}\langle \nabla h(w+\bar{w}), w^* - w\rangle. \end{aligned} \tag{18}$$

We divide (18) by $\alpha - 1$ and then sum it with (17), getting

$$h(w) + \frac{1}{\alpha-1}h(w^*) \ge h(w+\bar{w}) + \frac{1}{\alpha-1}h(w+\bar{w}).$$

Equivalently,

$$\frac{\alpha}{\alpha-1}(h(w+\bar{w}) - h(w^*)) \le h(w) - h(w^*).$$

So by rearranging,

$$h(w+\bar{w}) - h(w^*) \leq \left(1 - \frac{1}{\alpha}\right)\left(h(w) - h(w^*)\right) \,. \tag{19}$$

Thus we can equivalently write that

$$\psi^*(-w-\bar{w}) - \psi^*(-w^*) \leq \left(1 - \frac{1}{\alpha}\right)\left(\psi^*(-w) - \psi^*(-w^*)\right) \,. \tag{20}$$

Now, since by (15) we have $\frac{1}{\alpha}(B(F) - w) \subseteq B(G)$ and $w^* \in B(F)$, we have $\bar{w} = \frac{1}{\alpha}(w^* - w) \in B(G)$. Combining the fact that $\widetilde{w}$ is a minimizer of $\min\limits_{\widetilde{w}^* \in B(G)} \psi^*(-w - \widetilde{w}^*)$ (16) with the fact that $\bar{w} \in B(G)$, we have

$$\psi^*(-w') = \psi^*(-w - \widetilde{w}) \leq \psi^*(-w - \bar{w}) \,.$$

Combining this with (20), we obtain the desired claim:

$$\psi^*(-w') - \psi^*(-w^*) \leq \left(1 - \frac{1}{\alpha}\right)\left(\psi^*(-w) - \psi^*(-w^*)\right) \,. \tag{21}$$

The running time to compute the graphs G_i is $O\left(\sum\limits_{i=1}^{r} |V_i|^2 \mathcal{O}_i\right)$ and the time to run the algorithm from Lemma 4.3 is $\widetilde{O}\left(n + \sum\limits_{i=1}^{r} |V_i|^2\right)$, so the total running time is

$$\widetilde{O}\left(\sum_{i=1}^{r} |V_i|^2 \mathcal{O}_i + T_{\text{maxflow}}\left(n, n + \sum_{i=1}^{r} |V_i|^2\right)\right) \,.$$

□

Algorithm 4 Finding all minimum cuts

1: **function** FINDMINCUTS$(G(V,E,c), \phi, \varepsilon)$
2: $V' = V \cup \{s,t\}$, $E' = E \cup \bigcup\limits_{u \in V} \{(u,t)\} \cup \bigcup\limits_{u \in V} \{(s,u)\}$
3: Define parametric capacities

$$c_\lambda(u,v) = \begin{cases} \max\{0, \phi_u'(-\lambda)\} & \text{if } u \in V, v = t \\ \max\{0, -\phi_u'(-\lambda)\} & \text{if } u = s, v \in V \\ c_{uv} & \text{otherwise} \end{cases}$$

4: Set $(\Lambda, \tau) =$ APXPARAMETRICMINCUT$(G'(V',E'), c_\lambda, \lambda_{\min} = -(n+r)^{O(1)}, \lambda_{\max} = (n+r)^{O(1)})$
5: Set $\widetilde{w}_u = -\phi_u'^*(-\tau(u))$ for all $u \in V$
6: **return** $\widetilde{w}$