Selective Maximum Coverage and Set Packing

Felix J. L. Willamowski[1*] and Björn F. Tauer[2†]

[1] Lehrstuhl für Operations Research
RWTH Aachen University
willamowski@or.rwth-aachen.de

[2] Lehrstuhl für Management Science
RWTH Aachen University
tauer@algo.rwth-aachen.de

Abstract. In this paper we introduce the selective maximum coverage and the selective maximum set packing problem and variants of them. Both problems are strongly related to well studied problems such as maximum coverage, set packing, and (bipartite) hypergraph matching. The two problems are given by a collection of subsets of a ground set and index subsets of the indices of these subsets. Additionally, there are weights either for each element of the ground set or each subset for each index subset. The goal is to find at most one index per index subset such that the total weight of covered elements or of disjoint subsets is maximum. Applications arise in transportation, e.g., dispatching for ride-sharing services. We prove strong intractability results for the problems and provide almost best possible approximation guarantees.

1 Introduction and Preliminaries

We introduce the *selective maximum coverage* problem (SMC) generalizing the weighted maximum k coverage problem (WMKC) which is given by a finite ground set X with weights $w : X \to \mathbb{Q}_{\geq 0}$, a finite collection $\mathcal{S}$ of subsets of X, and an integer $k \in \mathbb{Z}_{\geq 0}$. The goal is to find a subcollection $\mathcal{S}' \subseteq \mathcal{S}$ containing at most k subsets and maximizing the total weight of covered elements $\sum_{x \in X'} w(x)$ with $X' = \cup_{S \in \mathcal{S}'} S$. For unit weights, $w \equiv 1$, the problem is called maximum k coverage problem (MKC).

With the SMC we have a collection of index subsets of the subsets of the ground set instead of a parameter k, where we can choose at most one index of each index subset. Formally, SMC is given by a finite ground set of elements $U = \{u_i\}_{i \in [n]}$ with weights $w : U \to \mathbb{Q}_{\geq 0}$, a finite collection of subsets $\mathcal{C} = \{C_i\}_{i \in [m]}$ of U, a finite index set $\mathcal{L}$, and a finite collection of index subsets of the subset indices $\{\mathcal{I}_\ell\}_{\ell \in \mathcal{L}}$, i.e., $\mathcal{I}_\ell \subseteq [m]$ for each $\ell \in \mathcal{L}$. The goal is to find an index subset $\mathcal{L}^* \subseteq \mathcal{L}$ and one index $i^*_\ell \in \mathcal{I}_\ell$ for each $\ell \in \mathcal{L}^*$ such that $\sum_{u \in C^*} w(u)$ is maximum, where $C^* = \cup_{\ell \in \mathcal{L}^*} C_{i^*_\ell}$ are the covered elements. See Figure 1 for a

*Corresponding author.

†We want to thank the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – UnRAVeL-Research Training Group 2236/1 for funding this research.

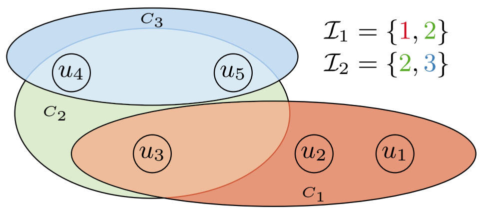

Fig. 1. SMC/SSP instance with five elements $U = \{u_1, \dots, u_5\}$ and three ground set subsets $C_1 = \{1, 2, 3\}$, $C_2 = \{3, 4, 5\}$, and $C_3 = \{4, 5\}$ and two index subsets $\mathcal{I}_1$ and $\mathcal{I}_2$.

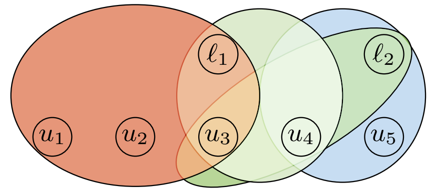

Fig. 2. Bipartite hypergraph matching problem instance with hypergraph $\mathcal{H} = (\mathcal{L} \dot\cup U, \mathcal{E})$, $|U| + |\mathcal{L}| = 7$ vertices, and $\sum_{\ell \in \mathcal{L}} |\mathcal{I}_\ell| = 4$ hyperedges.

visualization. We refer to the problem as *disjoint* (DSMC), if $\{C_{i^*_\ell}\}_{\ell \in \mathcal{L}^*}$ has to be pairwise disjoint. Note, that the special case of SMC with $\mathcal{I}_\ell \equiv [m]$ corresponds to a maximum coverage problem with $k = |\mathcal{L}|$.

The second problem we are considering is the *selective set packing* problem (SSP) generalizing the weighted set packing problem WSP. which is given by a finite ground set X and a finite collection of subsets $\mathcal{S}$ of X with weights $w : \mathcal{S} \to \mathbb{Q}_{\geq 0}$. The goal is to find a disjoint subcollection $\mathcal{S}' \subseteq \mathcal{S}$ of maximum total weight, i.e., each pair of subsets $S'_1 \neq S'_2 \in \mathcal{S}'$ is disjoint, $S'_1 \cap S'_2 = \emptyset$, and $\sum_{S \in S'} w(S)$ is maximum. In the case that each subset $S \in \mathcal{S}$ has size $k \in \mathbb{Z}_{\geq 0}$, i.e., $|S| = k$, the problem is called weighted k set packing problem (WKSP).

With the SSP we have (again) a collection of index subsets of the subsets of the ground set, where we can choose at most one index of each index subset. You can think of the SSP as DSMC only with a different objective function, instead of each element having a weight, all subsets have a weight that also depends on the index subset it was chosen from. Formally, we have weights $w_\ell : \mathcal{I}_\ell \to \mathbb{Q}_{\geq 0}$ for each subset and each index $\ell \in \mathcal{L}$ and the goal is to maximize $\sum_{\ell \in \mathcal{L}^*} w_\ell(i^*_\ell)$. See Figure 1 for a visualization. Note, that the special case of SSP with $\mathcal{L} = [m]$ and $\mathcal{I}_\ell = \{\ell\}$ for each $\ell \in \mathcal{L}$ corresponds to the weighted set packing problem.

Context The DSMC problem as well as the SSP problem can be applied during the controlling of a dynamic ride-sharing service, i.e., during the assignment process of passengers to vehicles. Especially the SSP setting models the assignment problem described in [1]. Within [1], the assignment problem is solved via an MILP-solver. The interpretation of the formal notation is as follows:
Each ground element $u \in U$ corresponds to a potential passenger of a ride-sharing service and the fleet of the operator consists of $\mathcal{L}$ vehicle. An upper bound on the size of C_i, $i \in [m]$, corresponds to a maximal passenger capacity $k \in \mathbb{Z}_{\geq 1}$ of each vehicle of the fleet. Depending on several constraints, the passenger pick up and desired drop off locations as well as the vehicle positions are analyzed and feasible tours are proposed. For our purpose it is sufficient to handle this feasible tour generator as black-box, hence we assume that the set of potential passenger vehicle combination is given. Accordingly, the collection of subsets $\mathcal{C}$ corresponds to the set of all feasible tours and the index subsets

$\{\mathcal{I}_\ell\}_{\ell\in\mathcal{L}}$ describe all feasible tours of vehicle $\ell \in \mathcal{L}$. Hence, except the fact that a passenger can only be assigned once, all other constraints are satisfied via the given input.

Both, DSMC and SSP cover different strategies of a ride-sharing operator. While DSMC corresponds to a revenue maximizing approach (e.g. each transported passenger pays her weight as service fee to the operator), the SSP version considers a profit maximizing approach. For example, $w_\ell(i_\ell^*)$ could involve the distances of all relevant locations of a tour $C_{i_\ell^*}$ as well as the initial position of vehicle ℓ and thus reduces the passenger dependent income by the corresponding travel costs.

1.1 Related Work

As already pointed out, most relevant to the results of this paper are the maximum covering problem as well as the weighted set packing problem, since DSMC (respectively SSP) are generalizations of those. Moreover, the problem to find a maximal matching in a hypergraph is connatural.

Maximum Covering The maximum covering problem, also called maximum coverage problem, is a widely studied problem. Due to the fact that already the unweighted version is NP-hard [6], approximation algorithms were investigated. It can be approximated within $(1-\frac{1}{e})$ via a classical greedy approach that selects always the set that covers the maximum number of uncovered elements until k sets are selected. Due to Feige, this greedy algorithm is essentially the best possible in terms of the approximation guarantee, since he proofed an inapproximation bound of $(1-\frac{1}{e}+\epsilon)$ [6].

Hypergraph Matching A bipartite matching is one of the fundamental tools for assignment processes in theory as well as in applications. The importance of this problem is underlined by various research in theoretical computer science to improve the running time of algorithms that compute a maximum matching in bipartite graphs within the last decades, e.g., see [14, 17, 19].

More related to our setup is the problem of computing a maximum matching in the hypergraph setting. An r-uniform bipartite hypergraph $\mathcal{H} = (X \dot\cup W, \mathcal{E})$ consists out of two disjoint vertex sets X and W and a set of hyperedges $\mathcal{E}$, where each edge contains exactly one vertex from X and $r-1$ vertices from W. Here, a (left-)perfect matching is a subset of disjoint edges such that each vertex in X is covered exactly ones. Annamalai proved the existence of an efficient algorithm to find these perfect matching whenever a stronger version of Haxell's condition holds [2]. Note that we can transform a maximum bipartite hypergrah matching instance $(\mathcal{H}, w)$ with $w : \mathcal{E} \to \mathbb{Q}_{\geq 0}$ into a SSP instance as follows. Set $U = W$, $\mathcal{L} = X$, $\mathcal{S} = \{S_i = e_i \cap X \mid e_i \in \mathcal{E}\}$, $\mathcal{I}_\ell = \{i \mid \ell \in e_i\}$ for each $\ell \in \mathcal{L}$, and $w_\ell(i) = w(e_i)$ for $\ell \in \mathcal{L}$ and $i \in \mathcal{I}_\ell$. The optimal value of the resulting SSP instance corresponds to the value of a maximum matching in $\mathcal{H}$. If all weights are set to one and the optimal value is $|X|$ the solution can be trivially transformed into a left-perfect matching of $\mathcal{H}$. See Figure 1 and Figure 2 for a comparison.

Set-Packing The general set packing problem is one of the classical NP-complete problems [8] and has been studied for decades. Due to Håstad and Zuckerman it is known that there is no $|X|^{\varepsilon-1/2}$-approximation algorithm, unless P = NP, where X denotes the finite ground set of elements.

There exist several approximation algorithm for k set packing based on local improvements [4, 5, 7, 10, 16, 20], where the best known result to date is a $3/(k+1+\epsilon)$-approximation due to Cygan. On contrary, there is no $\mathcal{O}(\ln(k)/k)$ to approximation algorithm, unless NP = P [13].

Note that one can transform a (weighted) set packing instance into a (weighted) independent set instance with $m = |\mathcal{S}|$ vertices [11]. Lehmann et al. showed that WSP admits a polynomial time $n^{-1/2}$-approximation, $n = |X|$, which is realized via a greedy approach [11, 18], while Gonen and Lehmann showed that no greedy algorithm can obtain a better solution.

To contrast DSMC and SSP with set packing it is worth to reconsider the underlying application. In contrast to WMKC and WSP, not every vehicle can perform every tour (due to its initial position), and hence the tours can not be selected arbitrary. On contrary, if all vehicles are placed at the same location, the index subsets of all vehicle would be identical and hence WMKC and WKSP are special cases of DSMC and SSP.

Observe that one can transform an instance of DSMC as well as an instance of SSP into an instance of the weighted set packing problem. Therefore, we define a new ground set $X' = U \cup \mathcal{L}$ and set the element weight $w(\ell) = 0$ for all $\ell \in \mathcal{L}$. For each subset $C_i \in \mathcal{C}$, $i \in [n]$, contained in $\mathcal{I}_\ell$, $\ell \in \mathcal{L}$, we create a new subset $S' = C \cup \{\ell\}$ and define the weight of this set equivalent to $\sum_{u \in S'} w(u)$ for the transformation of a DSMC instance and to $w_\ell(C_i)$ for SSP, respectively. Both described transformation are possible in polynomial time. Due to this transformation, one could solve DSMC and SSP via the methods known for WSP [11]. But due to the increase of the ground set (equally to the increase in Figure 2), the correspondent greedy approaches perform worse compared with the algorithms presented in this paper. The corresponding approximation factors applied on the transformed WSP instance are presented within Table 1.

1.2 Our contribution

Motivated by a ride-sharing application we present natural extensions of the weighted maximum k coverage problem as well as the weighted set packing problem. We prove strong intractability results for SMC, DSMC and SSP in Section 2 and provide almost best possible approximation guarantees in Section 3. The corresponding bounds are presented in Table 1.

Instance	**non-disjoint**	**disjoint**	
	SMC	DSMC	SSP
in-approximation guarantee	$(1-1/e+\varepsilon)$ Theorem 1	$\mathcal{O}(\ln(k)/k)$ $(k \in \mathbb{Z}_{\geq 3})$ Theorem 2	$n^{\varepsilon-1/2}$ Theorem 3
approximation factor greedy	$\frac{1}{2}$ Theorem 7	$\frac{1}{2\left(\sqrt{n}+1\right)}$ Algorithm 3	$\frac{1}{2\left(\sqrt{n}+1\right)}$ Algorithm 3
reformulation WSP	–	$\frac{1}{\left(\sqrt{n+\lvert L\rvert}\right)}$ [11, Theorem 3.3]	$\frac{1}{\left(\sqrt{n+\lvert L\rvert}\right)}$ [11, Theorem 3.3]

Table 1. Overview of in-/approximation results for SMC, DSMC and SSP. The first line states that it is hard to approximate the corresponding problem in polynomial time within a factor as denoted. The second line presents the approximation ratios of classic greedy approaches. Moreover, n denotes the number of elements of the ground set U, $\varepsilon > 0$ and e represents Euler's number.

2 Computational Complexity

In this section, we show the intractability of our problems. We show that there is no $(1-1/e+\varepsilon)$-approximation algorithm for SMC for every $\varepsilon > 0$ even if restricting to instances with unit weights, unless P = NP. As we pointed out, SMC is a generalization of MKC. We now prove this formally in the context of computational complexity by a reduction from MKC.

Theorem 1. *For any $\varepsilon > 0$, there is no $(1-1/e+\varepsilon)$-approximation algorithm for* SMC*, unless P = NP, even if restricting to instances with $\mathcal{I}_\ell = \{1, 2, \dots, m\}$ for $\ell \in \mathcal{L}$ and $w \equiv 1$.*

Proof. We reduce from MKC, which is known to be $(1-1/e+\varepsilon)$-hard to approximate for every $\varepsilon > 0$, unless NP = P [6]. Let $X = \{1, 2, \dots, q\}$ be the ground set, $\mathcal{S} = \{S_1, S_2, \dots, S_p\}$ be the collection of subsets, and $k \in \mathbb{Z}_{\geq 0}$ be the maximum number of subsets of a MKC instance. We construct an instance of SMC as follows. We create the ground set $U = X$, the subcollection of subsets $\mathcal{C} = \mathcal{S}$, $\mathcal{L} = \{1, 2, \dots, k\}$, $\mathcal{I}_\ell = \{1, 2, \dots, p\}$ for $\ell \in \mathcal{L}$, and $w \equiv 1$. Given a feasible solution of the MKC instance with subcollection $\mathcal{S}' = \{S_{j_i}\}_{i \in [k]}$, we create a feasible solution to the SMC instance as follows. Note that we can assume w.l.o.g. that the subcolection has exactly k subsets, since otherwise we could add subsets without decreasing the number of covered elements and still be feasible or if there are not more than $k' < k$ elements the solution is optimal and we could reduce k. We then set the index set to $\mathcal{L}^* = \{1, 2, \dots, k\}$, each index to $i^*_\ell = j_\ell$ $(\in \mathcal{I}_\ell)$ for $\ell \in \mathcal{L}^*$, Thus, we have a feasible solution to the SMC instance with value $|\cup_{\ell \in \mathcal{L}^*} C_{i^*_\ell}| = |\cup_{i \in [k]} S_{j_i}|$. Since this holds for every feasible solution of the MKC instance, we have $\text{OPT}_{\text{SMC}} \geq \text{OPT}_{\text{MKC}}$, where OPT_{MKC} and OPT_{SMC} denote the optimal values of the MKC and the SMC instance, respectively.

Let us assume that there is a $(1-1/e+\varepsilon)$-approximation algorithm for SMC for some $\varepsilon > 0$. We then get a solution with index subset $\mathcal{L}^*$ and indices i^*_ℓ for

$\ell \in \mathcal{L}^*$. This solution induces a solution to the MKC instance as follows. For $\ell \in \mathcal{L}^*$ add $S_{i^*_\ell}$ to the solution. Since, $|\mathcal{L}^*| \leq |\mathcal{L}| = k$ this is a feasible solution to the MKC instance. The number of covered elements is $|\cup_{\ell \in \mathcal{L}^*} S_{i^*_\ell}| = |\cup_{\ell \in \mathcal{L}^*} C_{i^*_\ell}|$, which is bounded from below by $(1 - 1/e + \varepsilon) \cdot \text{OPT}_{\text{SMC}}$. Then, the total number of covered elements in the constructed MKC solution is

$$|\bigcup_{\ell \in \mathcal{L}^*} S_{i^*_\ell}| = |\bigcup_{\ell \in \mathcal{L}^*} C_{i^*_\ell}| \geq (1 - 1/e + \varepsilon) \cdot \text{OPT}_{\text{SMC}} \geq (1 - 1/e + \varepsilon) \cdot \text{OPT}_{\text{MKC}}.$$

Thus, there is an $(1 - 1/e + \varepsilon)$-approximation algorithm for MKC, implying P = NP. It remains to note that the reduction is polynomial in the input size. □

We now show that the DSMC is even harder to approximate, i.e., there is no $\mathcal{O}(\ln(k)/k)$-approximation algorithm for every constant $k \in \mathbb{Z}_{\geq 3}$ for DSMC, unless P = NP. We prove this results by a reduction from WKSP.

Theorem 2. *For any constant $k \geq 3$, there is no $\mathcal{O}(\ln(k)/k)$-approximation algorithm for* DSMC*, unless P = NP, even if restricting to instances with $w \equiv 1$.*

Proof. We reduce from WKSP with unit weights, which is known to be $\mathcal{O}(\ln(k)/k)$-hard to approximate, unless NP = P [13]. Let $X = \{1, 2, \ldots, q\}$ be the ground set, $\mathcal{S} = \{S_1, S_2, \ldots, S_p\}$ be the collection of subsets, $w \equiv 1$ be the weights, and $k \in \mathbb{Z}_{\geq 3}$ be the integer of a WKSP instance. We construct an instance of DSMC as follows. We create the ground set $U = X$, the subcollection of subsets $\mathcal{C} = \mathcal{S}$, $\mathcal{L} = \{1, 2, \ldots, p\}$, $\mathcal{I}_\ell = \{1, 2, \ldots, p\}$ for $\ell \in \mathcal{L}$, and $w(u) = 1$ for each element $u \in U$. Given a feasible solution of the WKSP instance with subcollection $\mathcal{S}' = \{S_{j_i}\}_{i \in [p']}$ we create a feasible solution to the DSMC instance as follows. We set the index set to $\mathcal{L}^* = \{1, 2, \ldots, p'\}$, each index to $i^*_\ell = j_\ell$ $(\in \mathcal{I}_\ell)$ for $\ell \in \mathcal{L}^*$. Obviously, $\{C_{i^*_\ell}\}_{\ell \in \mathcal{L}^*}$ is pairwise disjoint. Thus, we have a feasible solution to the DSMC instance with cost $\sum_{e \in C^*} w(e) = k \cdot |\mathcal{S}'|$ with $C^* = \cup_{\ell \in \mathcal{L}^*} C_{i^*_\ell}$. Since this holds for every feasible solution of the WKSP instance, we have $\text{OPT}_{\text{DSMC}} \geq k \cdot \text{OPT}_{\text{WKSP}}$, where OPT_{WKSP} and OPT_{DSMC} denote the optimal values of the WKSP and the DSMC instance, respectively.

Let us assume that there is a $\mathcal{O}(\ln(k)/k)$-approximation algorithm for DSMC. We then get a solution with index subset $\mathcal{L}^*$, indices i^*_ℓ for $\ell \in \mathcal{L}^*$. This solution induces a solution to the WKSP instance as follows. For $\ell \in \mathcal{L}^*$ add $S_{i^*_\ell}$ to the solution. Since $\{C_{i^*_\ell}\}_{\ell \in \mathcal{L}^*}$ is pairwise disjoint, $\{S_{i^*_\ell}\}_{\ell \in \mathcal{L}^*}$ is also pairwise disjoint and this is a feasible solution to the WKSP instance. The number of sets of this WKSP solution is $|\mathcal{L}^*| = 1/k \cdot \sum_{\ell \in \mathcal{L}^*} w(C_{i^*_\ell})$. Then, the total number of covered elements in the constructed WKSP solution is

$$\frac{|\mathcal{L}^*|}{\text{OPT}_{\text{WKSP}}} = \frac{1/k \cdot \sum_{\ell \in \mathcal{L}^*} w(C_{i^*_\ell})}{\text{OPT}_{\text{WKSP}}} \geq \frac{\sum_{\ell \in \mathcal{L}^*} w(C_{i^*_\ell})}{\text{OPT}_{\text{DSMC}}} \in \mathcal{O}(\ln(k)/k).$$

Thus, there is an $\mathcal{O}(\ln(k)/k)$-approximation algorithm for WKSP, implying P = NP. It remains to note that the reduction is clearly polynomial. □

In the next theorem we show that there is no $n^{\varepsilon - 1/2}$-approximation algorithm for every $\varepsilon > 0$ for SSP by a reduction from the maximum independent set problem (MIS). The maximum independent set problem is given by an undirected graph $G = (V, E)$. The goal is to find a subset of nodes $S \subseteq V$ of maximum cardinality such that $\{u, v\} \notin E$ for each pair of node $u, v \in S$.

Theorem 3. *For any $\varepsilon > 0$, there is no $n^{\varepsilon-1/2}$-approximation algorithm for* SSP*, unless $P = NP$, even if restricting to instances with one index per index subset, $|\mathcal{I}_\ell| = 1$ for $\ell \in \mathcal{L}$, and weights $w_\ell(i) = 1$ for $\ell \in \mathcal{L}$ and $i \in \mathcal{I}_\ell$.*

Proof. We reduce from MIS, which is known to be $|V|^{\varepsilon-1}$-hard to approximate for every $\varepsilon > 0$, unless NP = P [15, 21], by a standard reduction from the *maximum clique* problem. Let $G = (V, E)$ be the graph of a MIS instance. We assume w.l.o.g. that each node is incident to at least one edge, i.e., $|\{v \in e \mid e \in E\}| \geq 1$ for each $v \in V$, since otherwise we could remove these nodes and solve the problem on the new instance and add them afterwards to the solution. We construct an instance of SSP as follows. We create the ground set $U = E$, for each $v \in V$ a subset $C_v = \{e \in E \mid v \in e\}$, the index subset $\mathcal{L} = V$, $\mathcal{I}_\ell = \{\ell\}$ for $\ell \in \mathcal{L}$, and $w_\ell(i_\ell) = 1$ for $\ell \in \mathcal{L}$ and $i_\ell \in \mathcal{I}_\ell$. Given a feasible solution of the MIS instance with nodes V', we create a feasible solution to the SSP instance as follows. We set the index set to $\mathcal{L}^* = V'$, and each index to $i^*_\ell = \ell$ $(\in \mathcal{I}_\ell)$ for $\ell \in \mathcal{L}^*$. The collection of subsets $\{C_{i^*_\ell}\}_{\ell \in \mathcal{L}^*}$ is pairwise disjoint, since otherwise there would be two sets C_{v_1} and C_{v_2} with $v_1 \neq v_2 \in V'$ sharing an edge. Thus, we have a feasible solution to the SSP instance with cost $\sum_{\ell \in \mathcal{L}^*} w_\ell(i^*_\ell) = |\mathcal{L}^*| = |V'|$. This holds for every feasible solution of the MIS instance, thus we have $\text{OPT}_{\text{SSP}} \geq \text{OPT}_{\text{MIS}}$, where OPT_{MIS} and OPT_{SSP} denote the optimal values of the MIS and the SSP instance, respectively.

Let us assume that there is a $n^{\varepsilon-1/2}$-approximation algorithm for SSP for some $\varepsilon > 0$. We then get a solution with index subset $\mathcal{L}^*$, and indices i^*_ℓ for $\ell \in \mathcal{L}^*$. This solution induces a solution to the MIS instance as follows. Since, $\{C_{i^*_\ell}\}_{\ell \in \mathcal{L}^*}$ is pairwise disjoint $V' = \mathcal{L}^*$ is a feasible solution to the MIS instance. Thus the number of vertices in this solution is $|V'| = \sum_{\ell \in \mathcal{L}^*} w_\ell(i^*_\ell)$. This is bounded from below by $n^{\varepsilon-1/2} \cdot \text{OPT}_{\text{SSP}}$. Then, the total number of vertices in the constructed MIS solution is

$$|V'| = \sum_{e \in C^*} w(e) \geq n^{\varepsilon-1/2} \cdot \text{OPT}_{\text{SSP}} \geq |V|^{\varepsilon'-1} \cdot \text{OPT}_{\text{MIS}}$$

with $\varepsilon' = 2 \cdot \varepsilon$. Thus, there is a $|V|^{\varepsilon'-1}$-approximation algorithm for MIS, implying P = NP. It remains to note that the reduction is clearly polynomial. □

Corollary 4. *For any $\varepsilon > 0$, there is no $m^{\varepsilon-1}$-approximation algorithm for* SSP*, unless $P = NP$, even if restricting to instances with one index per index subset, $|\mathcal{I}_\ell| = 1$ for $\ell \in \mathcal{L}$, and weights $w_\ell(i) = 1$ for $\ell \in \mathcal{L}$ and $i \in \mathcal{I}_\ell$.*

Proof. This is easy to see, since in the previous construction $|V| = |\mathcal{C}| = m$. □

We introduce a variant of SSP which is relevant in practice, the *selective set packing problem with lower sets* (SSPL). In this problem we can not only select given sets, but also each subset of these sets. Formally, we have weights for each subset $w_i^\ell : 2^{C_i} \to \mathbb{Q}_{\geq 0}$ for $i \in \mathcal{I}_\ell$ for $\ell \in \mathcal{L}$ and the goal is to find an index subset $\mathcal{L}^* \subseteq \mathcal{L}$, one index $i_\ell^* \in \mathcal{I}_\ell$ for each $\ell \in \mathcal{L}^*$, and subsets $C_\ell^* \subseteq C_{i_\ell^*}$ such that $\{C_\ell^*\}_{\ell \in \mathcal{L}^*}$ are pairwise disjoint, i.e., $C_\ell^* \cap C_{\bar{\ell}}^* = \emptyset$ for all $\ell \neq \bar{\ell} \in \mathcal{L}^*$, and $\sum_{\ell \in \mathcal{L}^*} w_{i_\ell^*}^\ell(C_\ell^*)$ is maximum. In the following theorem we show that this version is as hard as SSP.

Theorem 5. *If there is an $\alpha(n)$-approximation algorithm for* SSPL*, then there is an $\alpha(n)$-approximation algorithm for* SSP*.*

Proof. Let U be the ground set, $\mathcal{C}$ the collection of subsets, $\{\mathcal{I}_\ell\}_{\ell \in \mathcal{L}}$ the index subsets, and $w_\ell : \mathcal{I}_\ell \to \mathbb{Q}_{\geq 0}$ for $\ell \in \mathcal{L}$ the weights of a SSP instance. We construct an instance of SSPL as follows. We take the SSP instance as SSPL instance except that we create weights $w_i^\ell(C_i) = w_\ell(i)$ for $\ell \in \mathcal{L}$ and $i \in \mathcal{I}_\ell$ and $w_i^\ell(C) = 0$ for $\ell \in \mathcal{L}$, $i \in \mathcal{I}_\ell$, and $C \subsetneq C_i$. Each feasible solution of the SSP instance with index subset $\mathcal{L}^*$ and indices i_ℓ^* for $\ell \in \mathcal{L}^*$ directly transfers to a solution with the same objective value by setting $C_\ell^* = C_{i_\ell^*}$ for $\ell \in \mathcal{L}^*$. Thus we have $\text{OPT}_{\text{SSPL}} \geq \text{OPT}_{\text{SSP}}$, where OPT_{SSP} and OPT_{SSPL} denote the optimal values of the SSP and the SSPL instance, respectively.

Let us assume that there is an $\alpha(n)$-approximation algorithm for SSPL. We then get a solution with index subset $\mathcal{L}^*$, indices i_ℓ^* for $\ell \in \mathcal{L}^*$, and subsets $C_\ell^* \subseteq C_{i_\ell^*}$ for $\ell \in \mathcal{L}^*$. This solution induces a solution to the SSP instance as follows. Let $\mathcal{L}_+^* = \{\ell \in \mathcal{L}^* \mid w_{i_\ell^*}^\ell(C_\ell^*) > 0\}$. Since $\{C_\ell^*\}_{\ell \in \mathcal{L}_+^*}$ is pairwise disjoint and $C_\ell^* = C_{i_\ell^*}$ for $\ell \in \mathcal{L}_+^*$, we have that $(\mathcal{L}_+^*, \{i_\ell^*\}_{\ell \in \mathcal{L}_+^*})$ is a feasible solution to the SSP instance with objective value $\sum_{\ell \in \mathcal{L}_+^*} w_\ell(i_\ell^*) = \sum_{\ell \in \mathcal{L}^*} w_i^\ell(C_\ell^*)$ which is bounded from below by $\alpha(n) \cdot \text{OPT}_{\text{SSPL}}$. Thus we have a solution for SSP with objective value

$$\sum_{\ell \in \mathcal{L}_+^*} w_\ell(i_\ell^*) = \sum_{\ell \in \mathcal{L}^*} w_{i_\ell^*}^\ell(C_\ell^*) \geq \alpha(n) \cdot \text{OPT}_{\text{SSPL}} \geq \alpha(n) \cdot \text{OPT}_{\text{SSP}}.$$

It remains to note that the reduction is polynomial in the input size, since the weight function of the SSPL instance has only polynomial many non-zero values and so is polynomial-time computable. The weights for every subset are not stated explicitly, and thus the used approximation algorithm runs in time polynomial in the input size. □

Corollary 6. *For any $\varepsilon > 0$, there is no $n^{\varepsilon - 1/2}$-approximation algorithm for* SSPL*, unless $P = NP$, even if restricting to instances with one index per index subset, $|\mathcal{I}_\ell| = 1$ for $\ell \in \mathcal{L}$, and weights $w : 2^{C_i} \to \{0, 1\}$ for $\ell \in \mathcal{L}$ and $i \in \mathcal{I}_\ell$.*

Proof. Using Theorem 3 and Theorem 5 and following the constructions in the proofs, the result follows immediately. □

3 Approximation Algorithms

Within this chapter we consider various greedy approaches for our problem variants. We start with an algorithm for the SMC problem. Due to the fact that the collection of selected subsets has not to be pairwise disjoint, there exist two variants of this algorithm, one in which for each $\ell \in \mathcal{L}$ a subset is chosen and the other one that stops as soon as adding any new subset would not increase the target value. Note that independent of the improvement check both algorithm return the same target values. Thus we focus in our analysis to the one without improvement check, since here it holds $\mathcal{L}' = \mathcal{L} = \mathcal{L}^*$.

Algorithm 1: Greedy for SMC

```
1 I' ← ∅, L' ← ∅, C' ← ∅
2 while |L'| < |L| do
3     ℓ, i*_ℓ ← argmax_{ℓ∈L\L', i∈I_ℓ} w(C_i ∪ C')
4     I' ← I' ∪ {i*_ℓ}, L' ← L' ∪ {ℓ}, C' ← C' ∪ C_{i'_ℓ}
5 end
6 return (L', I')
```

Theorem 7. *Greedy for* SMC *is an* $\frac{1}{2}$*-approximation algorithm for* SMC.

Proof. Let $(\mathcal{L}^*, \{i^*_\ell\}_{\ell\in\mathcal{L}^*})$ be an optimal solution with $C^* = \cup_{\ell\in\mathcal{L}^*} C_{i^*_\ell}$ and $(\mathcal{L}', \{i'_\ell\}_{\ell\in\mathcal{L}'})$ the solution produced by the greedy algorithm with $C' = \cup_{\ell\in\mathcal{L}'} C_{i'_\ell}$. Furthermore, let $\{C_{i^*_{\ell_1}}, C_{i^*_{\ell_2}}, \ldots, C_{i^*_{\ell_p}}\}$ be a cardinality minimal collection of subsets of the optimal solution covering the elements not covered by the greedy algorithm, i.e., $C^* \setminus C'$. Furthermore, let C'_t be the elements covered after iteration $t \in \{1, 2, \ldots, |\mathcal{L}'|\}$ by the algorithm, and $t(\ell)$ the iteration in which a subset is selected for index $\ell \in \mathcal{L}'$. Since, $\mathcal{L}' = \mathcal{L}$, we have $\{\ell_1, \ell_2, \ldots, \ell_p\} \subseteq \mathcal{L}'$. Then,

$$w(C_{i'_{\ell_j}} \setminus C'_{t(\ell_j)-1}) = \max_{i\in\mathcal{I}_{\ell_j}} w(C_i \setminus C'_{t(\ell_j)-1}) \geq \max_{i\in\mathcal{I}_{\ell_j}} w(C_i \setminus C') \geq w(C_{i^*_{\ell_j}} \setminus C')$$

holds for every index $j \in [p]$. Therefore, we have

$$w(C') \geq \sum_{j\in[p]} w(C_{i'_{\ell_j}} \setminus C'_{t(\ell_j)}) \geq \sum_{j\in[p]} w(C_{i^*_{\ell_j}} \setminus C') = w(C^* \setminus C').$$

Due to non negative weights also $w(C') \geq w(C' \cap C^*)$ holds and thus we conclude

$$w(C') \geq \tfrac{1}{2} w(C^* \setminus C') + \tfrac{1}{2} w(C^* \cap C') = \tfrac{1}{2} w(C^*).$$

□

The analysis of the Greedy for SMC is tight. Consider the SMC instance given by the ground set $U = \{1, 2\}$, subsets $C_1 = \{1\}$ and $C_2 = \{2\}$, index set $\mathcal{L} = \{1, 2\}$, index subsets $\mathcal{I}_1 = \{1, 2\}$, $\mathcal{I}_2 = \{1\}$, and weights $w(1) = w(2) = 1$. Then the greedy algorithm (could) select $\mathcal{L}' = \{1, 2\}$, $i'_1 = 1$, and $i'_2 = 1$ with objective value 1, but 2 is optimal with $\mathcal{L}^* = \{1, 2\}$, $i^*_1 = 2$, and $i^*_2 = 1$.

We can overcome the difficulty that we select a subset from a "wrong" index subset, by computing a weighted maximum matching. We state this in the MatchingGreedy for SMC. Nevertheless, this algorithm does not give a better approximation ratio. Consider the SMC instance given by the ground set $U = \{1, 2, 3, 4\}$, subsets $C_1 = \{1, 2\}$, $C_2 = \{3, 4\}$, $C_3 = \{1\}$, $C_4 = \{2\}$, index set $\mathcal{L} = \{1, 2, 3\}$, index subsets $\mathcal{I}_1 = \{1, 2\}$, $\mathcal{I}_2 = \{3\}$, $\mathcal{I}_3 = \{4\}$, and weights $w \equiv 1$. Then, the MatchingGreedy for SMC (could) select $\mathcal{L}' = \{1, 2, 3\}$, $i'_1 = 1$, $i'_2 = 3$, and $i'_2 = 4$ with objective value 2, but 4 is optimal with $\mathcal{L}^* = \{1, 2, 3\}$, $i^*_1 = 2$, $i^*_2 = 3$, and $i^*_2 = 4$.

Algorithm 2: MatchingGreedy for SMC

$\mathcal{I}' \leftarrow \emptyset$ and $C' \leftarrow \emptyset$
while $|i \in [m] \mid \ell \in|$ **do**
  $V \leftarrow \mathcal{I}' \cup v_{\text{new}}$
  $E \leftarrow \{\{\ell, v\} \mid \ell \in \mathcal{L}, v \in V, v \in \mathcal{I}_\ell\}$
  $c(\ell, v_{\text{new}}) \leftarrow \max_{i \in \mathcal{I}_\ell \setminus \mathcal{I}'} w(C_i \setminus C') \quad \forall\, \ell \in \mathcal{L}$
  $c(e) \leftarrow 0 \quad \forall\, e \in E$ with $e \cap v_{\text{new}} = \emptyset$
  Compute a maximum weighted matching M on $(\mathcal{L} \cup V, E, c)$.
  if $c(M) = 0$ **or** $|M| \leq$ **then break**
  Let $\bar{\ell} \in \mathcal{L}$ matched with v_{new} in M and $i'_{\bar{\ell}} \in \mathcal{I}_{\bar{\ell}} \setminus \mathcal{I}'$ with $w(C_{i'_{\bar{\ell}}}) = c(M)$.
  $\mathcal{I}' \leftarrow \mathcal{I}' \cup \{i'_{\bar{\ell}}\}$
  $C' \leftarrow C' \cup C_{i'_{\bar{\ell}}}$
end
$\mathcal{L}' \leftarrow \mathcal{L} \cap M$
return $(\mathcal{L}', \mathcal{I}')$

The greedy algorithm for SSP, which is a modification of [11] and [12], initially removes all sets of cardinality $\sqrt{n}$ or more. Afterwards, indexes ℓ and i'_ℓ are greedily selected, such that $w_\ell(C_{i'_\ell})$ is maximum and ℓ as well as $C_{i'_\ell}$ are disjoint form the previous selected indexes/sets.

Theorem 8. *Greedy for* SSP *is an* $\frac{1}{2\sqrt{n}+1}$*-approximation algorithm for* SSP.

Proof. Let $(\mathcal{L}^*, \{i^*_\ell\}_{\ell \in \mathcal{L}^*})$ be an optimal solution with $C^* = \cup_{\ell \in \mathcal{L}^*} C_{i^*_\ell}$ and $(\mathcal{L}', \{i'_\ell\}_{\ell \in \mathcal{L}'})$ the solution produced by the greedy algorithm with $C' = \cup_{\ell \in \mathcal{L}'} C_{i'_\ell}$. Furthermore, for each iteration $t \in \{1, 2, \ldots, |\mathcal{L}'|\}$ let $\ell(t)$ be the index subset selected by the greedy algorithm and $\mathcal{L}(t) = \cup_{j=1}^t \ell(t)$. Consider the indices removed in some iteration t, $r(t) = \{i \mid \ell \in \mathcal{L} \setminus \mathcal{L}(t), i \in \mathcal{I}_\ell \setminus R(t-1), C_i \cap C_{i'_{\ell(t)}} \neq \emptyset\}$ and $R(t) = \cup_{j=1}^t r(t)$ with $R(0) = \emptyset$. Note that due to the pre-selection of small subsets of the greedy algorithm $|C_{i'_{\ell(t)}}| < \sqrt{n}$ holds and therefore $r(t)$ contains at most $\sqrt{n}$ indices of the optimal solution, because for $i \in R(t)$ the set C_i shares at least one unique element with $C_{i'_{\ell(t)}}$.

Algorithm 3: Greedy for SSP

$\mathcal{I}' \leftarrow \emptyset$, $\mathcal{L}' \leftarrow \emptyset$, $R \leftarrow \{i \in [m] \mid |C_i| \geq \sqrt{n}\}$
while $|\{i \in [m] \mid \ell \in \mathcal{L} \setminus \mathcal{L}', i \in \mathcal{I}_\ell \setminus R\}| \geq 1$ **do**
  $\ell, i'_\ell \leftarrow \operatorname*{argmax}_{\ell \in \mathcal{L} \setminus \mathcal{L}', i \in \mathcal{I}_\ell \setminus R} w_\ell(C_i)$
  $\mathcal{I}' \leftarrow \mathcal{I}' \cup \{i'_\ell\}$, $\mathcal{L}' \leftarrow \mathcal{L}' \cup \{\ell\}$, $R \leftarrow R \cup \{i \in [m] \mid C_i \cap C_{i'_\ell} \neq \emptyset\}$
end
$\bar{\ell}, \bar{i} \leftarrow \operatorname*{argmax}_{\ell \in \mathcal{L}, i \in \mathcal{I}_\ell, |C_i| \geq \sqrt{n}} w_\ell(C)$
if $w_{\bar{\ell}}(C_{\bar{i}}) \geq \sum_{\ell \in \mathcal{L}'} w_\ell(C_{i'_\ell})$ **then**
  return $(\{\bar{\ell}\}, \{\bar{i}\})$
else
  return $(\mathcal{L}', \mathcal{I}')$
end

Consider an arbitrary iteration t. The greedy approach may prevent optimal decisions for sets with at most $\sqrt{n}$ elements twofold. First, the selected subset overlaps with at most $\sqrt{n}$ optimal subsets $C_{i^*_\ell}$ with $i^*_\ell \in r(t)$. Second, there could exist an optimal set that does not belong to $\mathcal{L} \setminus \mathcal{L}(t)$. Thus, this subset is also no longer accessible, even if it is disjoint with all remaining subsets. If $\ell(t)$ is also selected in the optimal solution, we denote the corresponding index with $i^*_{\ell(t)}$. In the following estimation we consider those optimal indices only if $|C_{i^*_{\ell(t)}}| < \sqrt{n}$ and so we have

$$\begin{aligned}
\left(\sqrt{n}+1\right) \cdot w_{\ell(t)}(C_{i(t)}) &= \left(\sqrt{n}+1\right) \cdot \left(\max_{\ell \in \mathcal{L} \setminus \mathcal{L}(t-1), i \in \mathcal{I}_\ell \setminus R(t-1)} w_\ell(C_i) \right) \\
&\geq \sqrt{n} \cdot \left(\max_{\ell \in \mathcal{L} \setminus \mathcal{L}(t-1), i \in \mathcal{I}_\ell \setminus R(t-1)} w_\ell(C_i) \right) + w_{\ell(t)}(C_{i^*_{\ell(t)}}) \\
&\geq \sum_{\ell \in \mathcal{L}^*, i^*_\ell \in r(t)} w_\ell(C_{i^*_\ell}) + w_{\ell(t)}(C_{i^*_{\ell(t)}})
\end{aligned}$$

and thus

$$\begin{aligned}
\left(\sqrt{n}+1\right) \cdot \sum_{t \in \{1,2,\ldots,|\mathcal{L}'|\}} w_{\ell(t)}(C_{i(t)}) &\geq \sum_{t \in \{1,2,\ldots,|\mathcal{L}'|\}} \sum_{i^*_\ell \in r(t)} w_\ell(C_{i^*_\ell}) + w_{\ell(t)}(C_{i^*_{\ell(t)}}) \\
&\geq \sum_{\ell \in \mathcal{L}^*, |C_{i^*_\ell}| < \sqrt{n}} w_\ell(C_{i^*_\ell}).
\end{aligned}$$

Note that there are at most $\sqrt{n}$ optimal indices in $\{i \in [m] \mid |C_i| \geq \sqrt{n}\}$ and that the objective value of the greedy algorithm is at least as large as the value of the set with the largest weight. Hence

$$\sqrt{n} \cdot \sum_{t \in \{1,2,\ldots,|\mathcal{L}'|\}} w_{\ell(t)}(C_{i(t)}) \geq \sum_{\ell \in \mathcal{L}^*, |C_{i^*_\ell}| \geq \sqrt{n}} w_\ell(C_{i^*_\ell}).$$

Therefore we can conclude that the greedy solution has value of

$$\sum_{t \in \{1,2,\ldots,|\mathcal{L}'|\}} w_{\ell(t)}(C_{i(t)}) \geq \frac{1}{2\sqrt{n}+1} \sum_{\ell \in \mathcal{L}^*} w_\ell(C_{i^*_\ell})$$

□

Based on the similarities of our problem formulation to k set packing, we can benefit of the quasi polynomial time the $2/(k+1+\epsilon)$-approximation of Berman [3] for the weighted case and the $3/(k+1+\epsilon)$-approximation of Cygan [4] for the unweighted case and achieve also an approximation algorithm based on the maximal size of the subsets C_i, $i \in [m]$.

Corollary 9. *For any $\varepsilon > 0$, there exist a $2/(k+2+\epsilon)$-approximation algorithm for* SSP*, if $|C_i| \leq k$, $k \in \mathbb{Z}_{\geq 1}$, for all $i \in [m]$.*

Corollary 10. *For any $\varepsilon > 0$, there is a quasi polynomial time $3/(k+2+\epsilon)$-approximation for* SSP *with unit weights, if $|C_i| \leq k$, $k \in \mathbb{Z}_{\geq 1}$, for all $i \in [m]$.*

Proof. Reconsider the connection of SSP and set packing as introduced earlier. The presented transformation adds a single element from $\mathcal{L}$ to each copy of subset C_i. Thus, if the cardinality of each subset of the SSP is upper bounded by k, the cardinality of each subset in a k set packing is upper bounded by $k+1$. Note that the ground set of elements in this k set packing instance is also increased, but due to the fact that the k set packing approximation guarantee in [3] is independent of the cardinality of the ground set, we can apply the same algorithm and get the desired approximation guarantee also for SSP.

Bibliography

[1] Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., Rus, D.: On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. Proc. Natl. Acad. Sci. U.S.A. **114**(3), 462–467 (2017), doi:10.1073/pnas.1611675114, URL https://doi.org/10.1073/pnas.1611675114

[2] Annamalai, C.: Finding perfect matchings in bipartite hypergraphs. In: Krauthgamer, R. (ed.) Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016, pp. 1814–1823, SIAM (2016), ISBN 978-1-61197-433-1, doi:10.1137/1.9781611974331.ch126, URL https://doi.org/10.1137/1.9781611974331.ch126

[3] Berman, P.: A d/2 approximation for maximum weight independent set in d-claw free graphs. In: Algorithm Theory - SWAT 2000, pp. 214–219, Springer Berlin Heidelberg, Berlin, Heidelberg (2000), ISBN 978-3-540-44985-0

[4] Cygan, M.: Improved approximation for 3-dimensional matching via bounded pathwidth local search. In: 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, pp. 509–518, IEEE (2013)

[5] Cygan, M., Grandoni, F., Mastrolilli, M.: How to sell hyperedges: The hypermatching assignment problem. In: Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms, pp. 342–351, SIAM (2013)

[6] Feige, U.: A threshold of ln n for approximating set cover. J. ACM **45**(4), 634–652 (July 1998), ISSN 0004-5411, doi:10.1145/285055.285059, URL http://doi.acm.org/10.1145/285055.285059

[7] Fürer, M., Yu, H.: Approximating the k-set packing problem by local improvements. In: Fouilhoux, P., Gouveia, L.E.N., Mahjoub, A.R., Paschos, V.T. (eds.) Combinatorial Optimization - Third International Symposium, ISCO 2014, Lisbon, Portugal, March 5-7, 2014, Revised Selected Papers, Lecture Notes in Computer Science, vol. 8596, pp. 408–420, Springer (2014), ISBN 978-3-319-09173-0, doi:10.1007/978-3-319-09174-7_35, URL https://doi.org/10.1007/978-3-319-09174-7_35

[8] Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, NY, USA (1979)

[9] Gonen, R., Lehmann, D.: Optimal solutions for multi-unit combinatorial auctions: Branch and bound heuristics. CoRR **cs.GT/0202032** (2002), URL https://arxiv.org/abs/cs/0202032

[10] Halldórsson, M.M.: Approximating discrete collections via local improvements. In: SODA, vol. 95, pp. 160–169 (1995)

[11] Halldórsson, M.M.: Approximations of weighted independent set and hereditary subset problems. In: Asano, T., Imai, H., Lee, D.T., Nakano, S.i., Tokuyama, T. (eds.) Computing and Combinatorics, pp. 261–270, Springer Berlin Heidelberg, Berlin, Heidelberg (1999), ISBN 978-3-540-48686-2

[12] Halldórsson, M.M., Kratochvııl, J., Telle, J.A.: Independent sets with domination constraints. Discrete Applied Mathematics **99**(1), 39 – 54 (2000), ISSN 0166-218X, doi:https://doi.org/10.1016/S0166-218X(99)00124-9, URL http://www.sciencedirect.com/science/article/pii/S0166218X99001249

[13] Hazan, E., Safra, S., Schwartz, O.: On the complexity of approximating k-set packing. Comput. Complex. **15**(1), 20–39 (May 2006), ISSN 1016-3328, doi:10.1007/s00037-006-0205-6, URL https://doi.org/10.1007/s00037-006-0205-6

[14] Hopcroft, J.E., Karp, R.M.: An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. SIAM J. Comput. **2**(4), 225–231 (1973), doi:10.1137/0202019, URL https://doi.org/10.1137/0202019

[15] Håstad, J.: Clique is hard to approximate within $n^{1-\varepsilon}$. Acta Math. **182**(1), 105–142 (1999), doi:10.1007/BF02392825, URL https://doi.org/10.1007/BF02392825

[16] Hurkens, C.A.J., Schrijver, A.: On the size of systems of sets every t of which have an sdr, with an application to the worst-case ratio of heuristics for packing problems. SIAM Journal on Discrete Mathematics **2**(1), 68–72 (1989)

[17] Karzanov, A.V.: O nakhozhdenii maksimal'nogo potoka v setyakh spetsial'nogo vida i nekotorykh prilozheniyakh. Mathematicheskie Voprosy Upravleniya Proizvodstvom (1973)

[18] Lehmann, D., O'Callaghan, L., Shoham, Y.: Truth revelation in approximately efficient combinatorial auctions. J. ACM **49**(5), 577–602 (2002), doi:10.1145/585265.585266, URL https://doi.org/10.1145/585265.585266

[19] Madry, A.: Navigating central path with electrical flows: from flows to matchings, and back. CoRR **abs/1307.2205** (2013), URL http://arxiv.org/abs/1307.2205

[20] Sviridenko, M., Ward, J.: Large neighborhood local search for the maximum set packing problem. In: International Colloquium on Automata, Languages, and Programming, pp. 792–803, Springer (2013)

[21] Zuckerman, D.: Linear degree extractors and the inapproximability of max clique and chromatic number. In: Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing, p. 681–690, STOC '06, Association for Computing Machinery, New York, NY, USA (2006), ISBN 1595931341, doi:10.1145/1132516.1132612, URL https://doi.org/10.1145/1132516.1132612