

Assignment 2

1. **Non-oblivious Local Search for Max-2-SAT.** Fill in the details for the proof that the non-oblivious local search algorithm for max-2-sat obtains a $3/4$ approximation ratio. In particular, verify the “key lemma” stated on slide 4.

Lemma 1. *Let $P_{i,j}$ be the weight of all clauses in S_i containing x_j . Let $N_{i,j}$ be the weight of all clauses in S_i containing $\neg x_j$. For some local optimum $\hat{\tau}$ with respect to the potential function $3/2W(S_1) + 2W(S_2)$, for all variables x_j ,*

$$-\frac{1}{2}P_{2,j} - \frac{3}{2}P_{1,j} + \frac{1}{2}N_{1,j} + \frac{3}{2}N_{0,j} \leq 0.$$

Proof. Without loss of generality, suppose that the variables have been renamed such that all unnegated variables are set to true. Fix some variable x_j in some locally optimal truth assignment $\hat{\tau}$ with respect to $3/2W(S_1) + 2W(S_2)$. Let τ be some arbitrary assignment. Since $\hat{\tau}$ is locally optimal, flipping the truth value of x_j must not increase the potential or otherwise we would obtain a truth assignment with a greater potential than the current local optimal $\hat{\tau}$. Consider the clauses in $P_{1,j}$, $N_{0,j}$, $N_{1,j}$, and $P_{2,j}$. In the original truth assignment, precisely those in $P_{1,j}$ and $N_{1,j}$ are in S_1 and likewise, precisely those in $P_{2,j}$ is in S_2 . Note that because of the assumption that all unnegated variables are assigned to true, $N_{2,j}$ and $P_{0,j}$ are empty. After x_j is flipped:

- $N_{1,j}$ initially has one satisfied literal and contains $\neg x_j$. If x_j is flipped, the literal $\neg x_j$ is satisfied, along with the literal that is already satisfied. Hence, $N_{1,j} \subseteq S_2$ after we flip x_j .
- $N_{0,j}$ initially has no satisfied literal and contains $\neg x_j$. After x_j is flipped, the literal $\neg x_j$ is satisfied. Hence, $N_{0,j} \subseteq S_1$ after we flip x_j .
- $P_{1,j}$ initially has one satisfied literal x_j . After x_j is flipped, the only satisfied literal is no longer true. Hence $P_{1,j} \subseteq S_0$ after we flip x_j .
- $P_{2,j}$ initially has two satisfied literal including x_j . After x_j is flipped, x_j is no longer satisfied, leaving the other literal the only satisfied literal in each clause. Hence, $P_{2,j} \subseteq S_1$ after we flip x_j .

Now consider the change in the potential function as stated above. Since $\hat{\tau}$ is a local optimum with respect to the potential, the change in potential after flipping x_j must be non-positive.

$$\begin{aligned} \Phi(\tau') - \Phi(\tau) &= \frac{3}{2} (W(S'_1) - W(S_1)) + 2 (W(S'_2) - W(S_2)) \\ &= \frac{3}{2} (N_{0,j} + P_{2,j} - P_{1,j} - N_{1,j}) + 2 (N_{1,j} - P_{2,j}) \\ &= \frac{3}{2} N_{0,j} - \frac{1}{2} P_{2,j} - \frac{3}{2} P_{1,j} + \frac{1}{2} N_{1,j} \\ &\leq 0 \end{aligned}$$

as desired. □

2. Consider again the weighted d -set packing problem that we saw in terms of greedy algorithms. Show that the following oblivious local search algorithm obtains a $\frac{1}{d}$ approximation and that this bound is tight for this algorithm:

```

1  choose any initial feasible solution  $G$ 
2  for any input set  $S_i$ , let  $G_S = \{S_j \in G : S_i \cap S_j \neq \emptyset\}$ 
3  while  $\exists S. W(G \cup \{S\} \setminus G_S) > W(G)$ 
4       $G = G \setminus G_S \cup \{S\}$ 

```

Proof. We first prove that the local search algorithm indeed achieves a $1/d$ approximation using a charging argument. Let ALG denote some solution produced by the local search algorithm and OPT be some arbitrary optimal solution. Both ALG and OPT are feasible solutions to the d -set packing problem, so the sets in each of ALG and OPT are disjoint. We show that for any $S \in ALG$, the total weight of sets in OPT charged to S is at most $d \cdot W(S)$.

Fix some $S_{OPT} \in OPT$. If $S_{OPT} \in OPT \cap ALG$ (i.e. S_{OPT} is a set that is present in both the OPT solution and the local optimum), then we charge the weight of S_{OPT} to itself and the claim trivially holds. It then suffices to consider without loss of generality the case where $S_{OPT} \notin ALG$. For $S_{OPT} \in OPT$, consider $G_{S_{OPT}} = \{S_j \in ALG : S_{OPT} \cap S_j \neq \emptyset\}$. Clearly, $G_{S_{OPT}}$ is not empty because otherwise ALG would not have been a local optimum and the local search algorithm would have added S_{OPT} to ALG . Further, $|G_{S_{OPT}}| \leq d$ because $|S_{OPT}| \leq d$ and sets in ALG are disjoint.

For each $S_j \in G_{S_{OPT}} \subseteq ALG$, we charge it the weight

$$W(S_{OPT}) \cdot \frac{W(S_j)}{\sum_{S_i \in G_{S_{OPT}}} W(S_i)}.$$

Since S_{OPT} is not part of the local optimum ALG , exchanging S_{OPT} with $G_{S_{OPT}}$ will not increase the overall weight, so

$$W(S_{OPT}) \leq \sum_{S_i \in G_{S_{OPT}}} W(S_i).$$

This implies that

$$W(S_{OPT}) \cdot \frac{W(S_j)}{\sum_{S_i \in G_{S_{OPT}}} W(S_i)} \leq W(S_j) \quad (1)$$

By generalization, this inequality above holds for all S_{OPT} and $S_j \in G_{S_{OPT}}$.

Now for each arbitrary $S \in ALG$, it can be charged by at most d sets in OPT because it contains at most d elements and can intersect with at most d sets in OPT . Applying inequality (1) over all sets in OPT that intersects S , we show that the total weight charged to S by sets in OPT is at most $d \cdot W(S)$. Therefore, by the charging argument, the local search algorithm achieves a $1/d$ approximation.

To see that the ratio is tight, consider the following example. Let $U = \{1, \dots, d\}$ be the universe and define the following sets:

- $S_0 = \{1, \dots, d\}$ with $W(S_0) = d$
- $S_i = \{i\}$ with $W(S_i) = d$ for all $i \in \{1, \dots, d\}$

Let $G = \{S_0\}$ be the initial feasible solution. But then, as the algorithm is presented sets S_1 through S_d , the algorithm will not accept these sets as they all intersect with S_0 and exchanging S_0 for any of S_i with $i \in \{1, \dots, d\}$ does not increase the total weight. In the end, the algorithm returns $G = \{S_0\}$ as the solution without taking any additional sets. This gives a result of weight d . However, the optimal would discard the initial feasible solution and return $\{S_1, \dots, S_d\}$ as solution, giving us a total weight of d^2 . Then, clearly, $W(ALG)/W(OPT) = 1/d$. \square

Intuitively, the example showing that the $1/d$ ratio shows that our local search algorithm indeed suffer from a pathological initial solution that forces the algorithm into a local optimum without reaching the global optimum.

3. For the uncapacitated facility location problem (UFL) or the k -metric median problem, discuss:

(a) What you think might be a good way to define a non-oblivious local search algorithm.

We consider the UFL problem. In the problem, we are given (F, C, d, f) where F is the set of facilities, C is the set of clients to be served, d is the distance function, and f is the opening cost function. For the oblivious local search, the algorithm seeks to reduce the cost.

```

1   $F' =$  some feasible solution
2  while  $\exists f \subseteq F \setminus F'. \exists f' \subseteq F'$  such that  $|f| \leq 1, |f'| \leq 1$ , and  $\text{cost}(F') - \text{cost}(F' \cup f \setminus f')$ 
3       $F' = F' \cup f \setminus f'$ 

```

This, however, does not take into consideration the density of the facilities. Instead, consider the potential function:

$$\Phi(F) = \text{cost}(F) + \sum_{f \in F} \frac{\sum_{c \in C_f} d(c, f)}{|C_f|}$$

where $\text{cost}(F) = \sum_{j \in C} d(j, F) + \sum_{f_i \in F} f_i$ is the sum of distance from clients and the sum of opening costs, and

$$C_f = \{\text{all clients served by } f\} = \{c \in C : d(c, f) \leq d(c, f'), \forall f' \in F\}.$$

$d(j, F)$ is defined the same way as in class: $d(j, F) = \min_{f \in F} d(j, f)$. Intuitively, the second term of the potential function is a measure of *sparsity* of the clients served by each facility. Ideally, in an optimal solution, the average distance from the clients to the facility serving them should be small.

We can define the following non-oblivious local search algorithm with respect to the given potential function.

```

1   $F' =$  some feasible solution
2  while  $\exists f \subseteq F \setminus F'. \exists f' \subseteq F'$  such that  $|f| \leq 1, |f'| \leq 1$ , and  $\Phi(F') - \Phi(F' \cup f \setminus f')$ 
3       $F' = F' \cup f \setminus f'$ 

```

In each iteration, the algorithm can:

- remove a facility (when $f' \neq \emptyset$ and $f = \emptyset$)
- add a facility (when $f' = \emptyset$ and $f \neq \emptyset$)
- replace an existing facility with a new facility (when both f and f' are not empty)

The algorithm seeks to locally optimize the potential function such that

$$\Phi(F') > \Phi(F' \cup f \setminus f')$$

Intuitively, this means the new solution after replacing f' with f reduces the cost and is not too much sparser than the original F' .

(b) How would you define a greedy algorithm for obtaining an initial solution.¹

¹Kamal Jain, Mohammad Mahdian, Evangelos Markakis, Amin Saberi, and Vijay V. Vazirani. 2003. Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP. J. ACM 50, 6 (November 2003), 795-824.

```

1   $F' = \emptyset$ 
2  while  $C \neq \emptyset$ 
3      greedily find  $(i, C')$  for  $i \in F$  and  $C' \subseteq C$  such that  $\frac{f_i + \sum_{j \in C'} d(i, j)}{|C'|}$ 
4       $F' = F' \cup \{i\}$ 
5       $f_i = 0$ 
6       $C = C \setminus C'$ 

```

The objective function that this greedy algorithm tries to minimize is

$$\frac{f_i + \sum_{j \in C'} d(i, j)}{|C'|}.$$

For a “good” feasible solution, we should try to minimize the opening cost and the distance between each facility and the clients it serves. In addition, we would like to maximize the clients served by each facility (hence the denominator of $|C'|$). This is similar to the sparsity measure defined for the non-oblivious local search algorithm.

Moreover, at each iteration, if a new facility is added to the solution set, we set its opening cost to 0. That way, we can prioritize expanding/changing the service region of a facility that is already open over opening a new facility in order to avoid the opening cost. Also note that we do not remove a facility from F to give us the opportunity to refine the solution by moving/expanding an existing facility.