Correlation Clustering with Penalties and Approximating the Reordering Buffer Management Problem

Amjad Aboud

Technion - Computer Science Department - M.Sc. Thesis MSC-2008-07 - 2008

Correlation Clustering with Penalties and Approximating the Reordering Buffer Management Problem

Research Thesis

Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science

Amjad Aboud

Submitted to the Senate of the
Technion - Israel Institute of Technology

Shevat 5768 Haifa January 2008

The Research Thesis was Done Under the Supervision of Prof. Yuval Rabani in the Faculty of Computer Science.

I would like to deeply thank my advisor, Prof. Yuval Rabani, for his devoted support, professional help and guidance throughout this research.

I thank my family for their love, encouragement and support, and my friends for all the good times we had.

The generous financial help of the Technion is gratefully acknowledged.

Contents

Abstract **1**

Notations and Abbreviations **3**

1 Correlation Clustering with Penalties **5**
1.1 Introduction 5
1.1.1 Previous Results 6
1.1.2 Our Results 7
1.2 The Algorithm 8
1.3 Analysis 11
1.4 The Weighted Case 14
1.5 Open Problems 16

2 Approximating the Reordering Buffer Management Problem **17**
2.1 Introduction 17
2.1.1 The Model 17
2.1.2 Previous Results 17
2.1.3 Our Contribution 18
2.2 Proof of the $\Omega(\log k)$ gap 19
2.3 Open Problems 23

Bibliography **26**

Abstract

Correlation Clustering with Penalties: We study a graph-theoretic clustering criterion that allows for the presence of outliers. More specifically, we consider the correlation clustering problem where, in addition to the input complete graph with edges marked "+" or "−", there is a penalty function on the nodes, and nodes can be discarded from the clustering at the cost of their assigned penalty. We give a simple 9-approximation algorithm for this problem. Our algorithm is based on the primal-dual schema. We use the primal-dual schema to compute a potentially infeasible solution whose purpose is to indicate an approximately best set of nodes to be discarded from the clustering. In addition, we consider a general version of the previous problem, where instead of marking edges with "+" or "−", we have two weight functions on the edges. A "+"-function ("−"-function) that indicates the desire of each edge to be marked "+" ("−"). We present the modification to the algorithm to solve the general version under specific probability, that achieves 17-approximation.

Reordering Buffer Management Problem: A sequence of objects which are characterized by their color has to be processed and their processing order influences how efficiently they can be processed. A color change between two successive objects produces non-uniform cost; this is the non-uniform case of the problem. A reordering buffer which is a random access buffer with storage capacity of k objects can be used to rearrange this sequence in such a way that the total cost is minimized. The strategy with the best known competitive ratio is MAP. An upper bound of $O(\log k)$ on the competitive ratio of MAP is known and a non-constant lower bound on the competitive ratio is not known [12]. Based on a specific sequence input, we proof that the previously used proof techniques are not suitable to show an $o(\log k)$ upper bound on the competitive ratio of MAP.

Notation and Abbreviations

V	–	Set of vertices.
E	–	Set of edges.
G	–	Graph.
C	–	Group of vertices denotes a cluster.
U	–	Group of vertices denotes outliers group.
T	–	Group of triples of vertices denotes Set of inconsistent triangles.
LP	–	Linear Program.
DLP	–	Dual linear Program.
$\mathbb{N}$	–	Natural numbers including the zero.
$PTAS$	–	Polynomial-time approximation scheme.
NP	–	Non-deterministic Polynomial-time.
K_n	–	Complete graph on n vertices.
RBM	–	Reordering Buffer Management.
$k-buffer$	–	Reordering buffer of size k.
b_c	–	Cost of changing to color c.
MAP	–	Maximal Adjusted Penalty.
P_c	–	Penalty counter for color c.
OPT_k	–	Optimal offline strategy with $k-buffer$.
α	–	$\left\lceil \log(\frac{k}{\log k}) \right\rceil$.
$(c_l)^m$	–	A package of m objects of color c_l.

Chapter 1

Correlation Clustering with Penalties

1.1 Introduction

Rigorous formulations of the clustering problem often are sensitive to the presence of outliers, arbitrary noisy data that spoils an otherwise clean partition of the data set. Some approximation algorithms for clustering data in high dimensional real space, under various objective functions, can be modified to handle outliers quite well, mostly in the form of allowing the removal of a fixed fraction of the points (see, for example [19, 2, 3, 8]). However, such results are rare for attractive graph-theoretic clustering models. In this thesis we study a clustering criterion that allows for the presence of outliers. The criterion is the *prize collecting* model that was previously studied in the context of network design problems (see [14]) and facility location problems [7]. The data set is endowed with a penalty function, the output clustering may cover any subset of the data set, and the clustering objective is to minimize the sum of a standard objective function on the clustered data, plus the total penalty on the unclustered data. Node penalties may model, for instance, the degree of confidence we have in each data point. Higher confidence translates into a higher penalty for omitting the point from the clustering.

Previously, Charikar et al. [7] examined the prize collecting criterion for facility location problems. They gave a 3-approximation algorithm for prize-collecting facility location, and a 4-approximation algorithm for prize-collecting k-median. Hassin and Or [15] recently studied this criterion in

the context of clustering by edge-cost minimization: Given a complete graph $G = (V, E)$, a weight function $w : E \to \mathbb{N}$, on its edges, and a penalty function $c : V \to \mathbb{N}$ on the vertices, the objective is to partition V into k clusters, $\{S_1, \ldots, S_k\}$ and a set of outliers U, as to minimize the sum of edges weights on the multi-cut (that cross the clusters) plus the cost of outliers. They gave a 2-approximation algorithm for the single cluster case, a PTAS for single cluster metric instances with uniform penalties, and a 2-approximation for metric instances with a constant number of clusters. (If outliers are not allowed, there is a PTAS for metric instances with a constant number of clusters [16, 8].)

We investigate the prize collecting criterion in the context of correlation clustering, a graph-theoretic formulation that was first studied by Bansal, Blum, and Chawla [4]. The original problem is stated as follows. We have a complete graph where the edges are marked either by "+" or by "−", indicating an inclination to be in the same cluster or in separate clusters, respectively. The goal is to partition the node set into clusters that respect these inclinations as best as possible. One of the attractions of this clustering formulation is that it does not require the input to specify in advance the number of clusters desired. Rather, the choice of this number is guided by the cost function. Clearly, we can state the clustering objective in two ways. Either we want to minimize the total number of edges that are marked inconsistently with the clustering (i.e., edges marked "+" in with endpoints in different clusters and edges marked "−" with endpoints in the same cluster), or we want to maximize the number of edges that are marked consistent with the clustering. Both objectives are equivalent at the optimum; the problem is NP-hard. They may (and they do) differ as far as approximation is concerned. As dictated by our outliers model, in this thesis we are concerned with the minimization version of the problem.

1.1.1 Previous Results

A succession of papers gave approximation algorithms for both versions of the problem (as well as its generalized forms where only a subset of the edges are marked) [4, 6, 9, 10, 20, 1, 13]. The first constant approximation for the minimization version on complete graphs was given by Bansal, Blum and Chawla [4]. This factor was improved to 4 by rounding a linear program by Charikar, Guruswami and Wirth [6]. The best result for the minimization version is the 3-approximation randomized greedy algorithm of Ailon,

Charikar, and Newman [1].

$CC - Pivot(G = (V, E_+))$

- Pick random vertex $v \in V$.
- Set $C = \{v\}, V' = \emptyset$.
- For all $u \in V, u \neq v$:
  - If $(v, u) \in E_+$
    - Add u to C
  - Else $((v, u) \in E_-)$
    - Add u to V'
- Let G' be the subgraph induced by V'.
- Return clustering $C, CC - Pivot(G')$.

E_+: The set of edges with "+" relation. E_-: The set of edges with "−" relation.

1.1.2 Our Results

Our algorithm uses the primal-dual schema, based on a linear programming relaxation. Our relaxation is a modification of the relaxation underlying the Ailon-Charikar-Newman dual fitting analysis of their algorithm.

The following is the relaxation used in [1].

$$\begin{array}{rll} \min & \sum_{e \in E_+ \cup E_-} y_e & \\ \text{subject to} & \sum_{e \subset t} y_e \geq 1 & \forall t \in T, \\ & y_e \geq 0 & \forall e \in E_+ \cup E_-. \end{array}$$

T denote the set of inconsistent triangles in G, i.e. the set of triples $\{x1, x2, x3\}$ which has two "+" relations and one "−" relation among them.

This is all fairly standard for solving prize collecting problems. For example, the prize collecting facility location algorithms in [7] are also based on the primal-dual schema. The interesting twist in our algorithm is that the primal-dual schema is used solely for the purpose of identifying the set of nodes that will be removed at the cost of paying their cumulative penalties. In fact, this phase of the algorithm does not even produce a feasible clustering solution. (The linear programming relaxation that we use has integral feasible solutions that do not correspond to any PRIZE COLLECTING CORRELATION CLUSTERING feasible solution; the same is true of the relaxation in [1].) We note that identifying the best set of nodes to remove is an NP-hard problem. After removing the set of nodes indicated by the first phase of the algorithm, a second phase runs the Ailon-Charikar-Newman 3-approximation algorithm on the remaining nodes, producing the output clustering. The total cost due to the clustering cost plus the penalties cost turns out to be a 9-approximation for PRIZE COLLECTING CORRELATION CLUSTERING.

The result of the thesis is organized as follows. In section 1.2 we present the algorithm, and in section 1.3 we analyze its performance. In section 1.4 we present the modification to the algorithm that solves the weighted version.

1.2 The Algorithm

In this section we describe our approximation algorithm, and in the next section we analyze its performance. Let $G = (V, E_+; c)$ denote an instance of PRIZE COLLECTING CORRELATION CLUSTERING, where V is the node set to be clustered, E_+ is the edges of the complete graph over V that are labeled "+" (so the other edges are labeled "−"; we denote them by E_-), and $c : V \to \mathbb{N}$ is the penalty function. Our goal is to compute a clustering $C_1, C_2, C_3, \ldots, C_k, U \subset V$, where k is an arbitrary positive integer, all the sets are mutually disjoint, their union is V. We want to minimize the cost of this solution, which is

$$|\{e \in E_+ : \ \forall i, \ |e \cap C_i| \leq 1\}| + |\{e \in E_- : \ \exists i, e \subset C_i\}| + \sum_{x \in U} c(x).$$

The algorithm proceeds in two phases. In phase I, we identify a set $U \subset V$ of nodes that are removed from the instance, adding $\sum_{x \in U} c(x)$ to the cost of the solution. In phase II, we run the pivoting algorithm from [1]

on $G' = (V \setminus U, E'_+)$, where E'_+ is the restriction of E_+ to $V \setminus U$. Let $z(G), z_{\text{opt}}(G), z(G'), z_{\text{opt}}(G')$ denote, respectively, the cost of our algorithm's solution for G, the cost of an optimal solution for G, the cost of the pivoting algorithm's solution for G', and the cost of an optimal solution for G', respectively. Then, we are guaranteed that $\mathrm{E}[z(G')] \leq 3z_{\text{opt}}(G')$, and thus $\mathrm{E}[z(G)] \leq 3z_{\text{opt}}(G') + \sum_{x \in U} c(x)$.

We now describe in detail phase I of the algorithm. Let T denote the set of inconsistent triangles in G, i.e.

$$T = \{\{x_1, x_2, x_3\} : \ x_1, x_2, x_3 \in V \wedge |\{\{x_1, x_2\}, \{x_1, x_3\}, \{x_2, x_3\}\} \cap E_+| = 2\}.$$

Consider the following linear program.

$$\begin{array}{rlll} z^*(G) = \min & \sum_{e \in E_+ \cup E_-} y_e + \sum_{x \in V} c(x) \cdot p_x & & \\ \text{subject to} & \sum_{e \subset t} y_e + \sum_{x \in t} p_x \geq 1 & \forall t \in T, & (1.1) \\ & y_e \geq 0 & \forall e \in E_+ \cup E_-, & \\ & p_x \geq 0 & \forall x \in V. & \end{array}$$

Let $C_1, C_2, \ldots, C_k, U$ be a feasible solution to PRIZE COLLECTING CORRELATION CLUSTERING (U is the set of outliers). Consider the following $\{0,1\}$ assignment to y, p. If $e = \{x_1, x_2\} \in E_+$ then set $y_e = 1$ iff there exist $i, j \in \{1, 2, \ldots, k\}$, $i \neq j$, such that $x_1 \in C_i$ and $x_2 \in C_j$. If $e = \{x_1, x_2\} \in E_-$ then set $y_e = 1$ iff there exists $1 \leq i \leq k$ such that $x_1, x_2 \in C_i$. Set $p_x = 1$ iff $x \in U$. Then, the vectors y, p are non-negative and satisfy constraints (1.1), and furthermore $\sum_{e \in E_+ \cup E_-} y_e + \sum_{x \in V} c(x) \cdot p_x$ is precisely the cost of the solution $C_1, C_2, \ldots, C_k, U$. Therefore, the above linear program is a relaxation of PRIZE COLLECTING CORRELATION CLUSTERING. Notice that the set of feasible solutions of this relaxation includes integral solutions that are not feasible solutions for the original problem.

The dual program is

$$\begin{array}{rlll} \text{maximize} & \sum_{t \in T} r_t & & \\ \text{subject to} & \sum_{t \supset e} r_t \leq 1 & \forall e \in E_+ \cup E_-, & (1.2) \\ & \sum_{t \ni x} r_t \leq c(x) & \forall x \in V, & (1.3) \\ & r_t \geq 0 & \forall t \in T. & \end{array}$$

We use the pair of dual programs to define a primal-dual algorithm that identifies a set of outlier nodes $U \subset V$. The variables p_x, $x \in U$, will be set

to 1. The algorithm also maintains a set of edges $E' \subset E_+ \cup E_-$ that will indicate which variables y_e should be set to 1. The algorithm raises active dual variables r_t (initially all dual variables are active) until a constraint (1.2) or (1.3) becomes tight. When that happens, an edge is added to E' or a node is added to U, respectively, and some dual variables are inactivated. When all variables become inactive, E' is cleaned up, and the algorithm terminates.

1. (Initiation) $y \leftarrow 0$, $p \leftarrow 0$, $r \leftarrow 0$, $U \leftarrow \emptyset$, $E' \leftarrow \emptyset$.
2. Uniformly raise active dual variables until a constraint (1.2) or a constraint (1.3) become tight.
3. For every edge $e \in E_+ \cup E_- \setminus E'$ such that $\sum_{t \supset e} r_t = 1$, add e to E' and inactivate the variables r_t for all $t \supset e$.
4. For every node $x \in V \setminus U$ such that $\sum_{t \ni x} r_t = c(x)$, add x to U and inactivate the variables r_t for all $t \ni x$.
5. While there exist active dual variables r_t go back to step 2.
6. Remove from E' all the edges e with $e \cap U \neq \emptyset$.
7. Put $\forall e \in E'$, $y_e \leftarrow 1$. Put $\forall x \in U$, $p_x \leftarrow 1$.
8. Output y, p, U, E'.

1.3 Analysis

Let $C_1, C_2, \ldots, C_k, U$ be a feasible solution to PRIZE COLLECTING CORRELATION CLUSTERING. Consider the corresponding $\{0,1\}$ assignment to y, p that we defined in the previous section.

Lemma 1. *The vectors y, p are non-negative and satisfy the constraints* (1.1).

Proof. Consider an inconsistent triangle $t \in T$. We do a case analysis.
Case 1: $t \cap U \neq \emptyset$. Let $x_t \in t \cap U$. Then, $p_{x_t} = 1$, so constraint (1.1) is satisfied.
Case 2: $t \cap U = \emptyset$ and for every $i \in \{1, 2, \ldots, k\}$, $|t \cap C_i| \leq 2$. Pick i such that $|t \cap C_i| = 1$, let $\{x_1\} = t \cap C_i$, and let $t = \{x_1, x_2, x_3\}$. Then, either $\{x_1, x_2\}$ or $\{x_1, x_3\}$ is marked "+". Denote this edge by e_t^+. In this case $y_{e_t^+} = 1$, so the constraint (1.1) is satisfied.
Case 3: $t \cap U = \emptyset$ and there exists $i \in \{1, 2, \ldots, k\}$ such that $t \subset C_i$. Let $e_t^- \subset t$ be the unique edge in t marked "$-$". As $e_t^- \subset C_i$, $y_{e_t^-} = 1$, so the constraint (1.1) is satisfied. □

Define:

$$\begin{aligned} a(y) &:= \sum_{e \in E_+ \cup E_-} y_e \\ b(p) &:= \sum_{x \in V} c(x) \cdot p_x \end{aligned}$$

Notice that for a given feasible solution for $G = (V, E_+; c)$ and its corresponding $\{0,1\}$ assignment to the vectors y, p, the cost of the given solution is exactly $a(y) + b(p)$, as

$$\begin{aligned} & \text{cost}(C_1, C_2, \ldots, C_k, U) \\ = \; & |\{e \in E_+ : \; \forall i, \; |e \cap C_i| \leq 1\}| + |\{e \in E_- : \; \exists i, e \subset C_i\}| + \sum_{x \in U} c(x) \\ = \; & a(y) + b(p). \end{aligned}$$

Let y, p, U, E' be the output of phase I of the algorithm from the previous section (the primal-dual phase).

Lemma 2. *The vectors y, p are an integral feasible solution of the linear program, and $a(y) + b(p) \leq 3 \cdot z^*(G)$.*

Proof. By step 7 of the algorithm, $a(y) = \sum_{e \in E'} y_e$ and $b(p) = \sum_{x \in U} c(x) \cdot p_x$. Also, by step 3 of the algorithm, if $e \in E'$ (so $y_e = 1$) then $\sum_{t \supset e} r_t = 1$. By step 4 of the algorithm, if $x \in U$ (so $p_x = 1$) then $\sum_{t \ni x} r_t = c(x)$. Finally, step 6 of the algorithm guarantees that for every $t \in T$, $\sum_{e \subset t} y_e + \sum_{x \in t} p_x \leq 3$. Therefore,

$$\begin{aligned} a(y) + b(p) &= \sum_{e \in E'} y_e + \sum_{x \in U} c(x) \cdot p_x \\ &= \sum_{e \in E'} \left(\sum_{t \supset e} r_t \right) \cdot y_e + \sum_{x \in U} \left(\sum_{t \ni x} r_t \right) \cdot p_x \\ &= \sum_{t \in T} r_t \cdot \left(\sum_{e \subset t} y_e + \sum_{x \in t} p_x \right) \\ &\leq 3 \cdot \sum_{t \in T} r_t \\ &\leq 3 \cdot z^*(G). \end{aligned}$$

This completes the proof. □

Theorem 3. $\mathrm{E}[z(G)] \leq 9 \cdot z_{\mathrm{opt}}(G)$.

Proof. Let T' denote the set of inconsistent triangles in G', and let

$$z^*(G') = \min \left\{ \sum_{e \in E'_+ \cup E'_-} y_e : \ \forall t \in T', \ \sum_{e \subset t} y_e \geq 1 \wedge y \geq 0 \right\}. \tag{1.4}$$

The pivoting algorithm of [1] guarantees that $E[z(G')] \leq 3 \cdot z^*(G')$. We

will show that $z^*(G') \leq a(y)$. This implies the theorem, as

$$\begin{aligned} E[z(G)] &= E[z(G')] + \sum_{x \in U} c(x) \\ &\leq 3 \cdot z^*(G') + \sum_{x \in U} c(x) \\ &\leq 3 \cdot a(y) + \sum_{x \in V} c(x) \cdot p_x \\ &\leq 3 \cdot (a(y) + b(p)) \\ &\leq 9 \cdot z^*(G) \\ &\leq 9 \cdot z_{\text{opt}}(G), \end{aligned}$$

where the penultimate inequality follows from Lemma 2, and the last inequality follows from the fact that the linear program relaxes PRIZE COLLECTING CORRELATION CLUSTERING.

In order to prove that $z_{\text{opt}}(G') \leq a(y)$, it is sufficient to show that y is a feasible solution for the right-hand side linear program in (1.4). Notice that for every inconsistent triangle $t = \{x_1, x_2, x_3\}$ such that $t \cap U = \emptyset$, we have $p_{x_1} = p_{x_2} = p_{x_3} = 0$. Thus, the constraint (1.1) of $\sum_{e \subset t} y_e + \sum_{x \in t} p_x \geq 1$ implies that $\sum_{e \subset t} y_e \geq 1$, so y satisfies the constraints in (1.4), as required. □

1.4 The Weighted Case

The problem: We are given a set of items V. a penalty function $c : V \to \mathbb{N}$, and two weight functions $w^+ : \binom{V}{2} \to \mathbb{N}$ and $w^- : \binom{V}{2} \to \mathbb{N}$. We must partition V into clusters $C_1, C_2, \ldots, C_k$, for an arbitrary positive integer k, and an outlier set U, minimizing the cost of the clustering. The cost of the clustering is

$$\sum_{1 \le i < j \le k} \sum_{e \in C_i \times C_j} w^+(e) + \sum_{i=1}^{k} \sum_{e \in \binom{C_i}{2}} w^-(e) + \sum_{x \in U} c(x).$$

In [1] it was shown how to get constant factor approximations for three special cases of WEIGHTED COLLECTING CORRELATION CLUSTERING. They gave a 5-approximation in the case of probability constraints, where $w^+(e) + w^-(e) = 1$ for all $e \in \binom{V}{2}$. They gave a 2-approximation in the case of probability constraints and triangle inequality constraints, where $w^-(\{i,k\}) \le w^-(\{i,j\}) + w^-(\{j,k\})$, for all $i, j, k \in V$.

We use those results and our method to get a 17-approximation algorithm for PRIZE COLLECTING WEIGHTED CORRELATION CLUSTERING, in the case of probability constraints. We need to show how to implement phase I in which we choose the outliers set, and then we can run phase II using the 5-approximation algorithm of [1].

We create an unweighted PRIZE COLLECTING CORRELATION CLUSTERING instance $G_w = (V, E_w^+; c)$, where $E_w^+ = \{e | w^+(e) \ge \frac{1}{2}\}$ and $E_w^- = \{e | w^-(e) = 1 - w^+(e) > \frac{1}{2}\}$. Define $w : E_w^+ \cup E_w^- \to \mathbb{N}$ as follows. If $e \in E_w^+$ then $w(e) = w^+(e)$. Otherwise ($e \in E_w^-$), $w(e) = w^-(e)$. Put $\overline{w}(e) = 1 - w(e)$. Also put $\widetilde{w}(e) = w(e) - \overline{w}(e)$ and $\widetilde{c}(x) = c(x) - \sum_{e \ni x} \overline{w}(e)$.

The primal and dual linear programming relaxations are modified as follows. The primal program we use is:

$$\begin{aligned} h^*(G_w) = \min \quad & \textstyle\sum_{e \in E_w^+ \cup E_w^-} \widetilde{w}(e) \cdot y_e + \sum_{x \in V} \widetilde{c}(x) \cdot p_x & \\ \text{subject to} \quad & \textstyle\sum_{e \subset t} y_e + \sum_{x \in t} p_x \ge 1 & \forall t \in T, \\ & y_e \ge 0 & \forall e \in E_w^+ \cup E_w^-, \\ & p_x \ge 0 & \forall x \in V. \end{aligned}$$

It's dual program is:

$$\begin{array}{rll} \text{maximize} & \sum_{t \in T} r_t & \\ \text{subject to} & \sum_{t \supset e} r_t \leq \widetilde{w}(e) & \forall e \in E_w^+ \cup E_w^-, \\ & \sum_{t \ni x} r_t \leq \widetilde{c}(x) & \forall x \in V, \\ & r_t \geq 0 & \forall t \in T. \end{array}$$

The quantity $h^*(G_w) + \sum_{e \in E_w^+ \cup E_w^-} \overline{w}(e)$ is a lower bound on the value of an optimal solution for the problem. Notice that $\widetilde{w}(e) \geq 0$ for all $e \in E_w^+ \cup E_w^-$. If for all $x \in V$ we can assume that $\widetilde{c}(x) \geq 0$, then our previous Phase I and II algorithm gives a 15-approximation guarantee, using a similar analysis. However, the latter assumption may be false. In order to fix this problem, we execute a preliminary Phase 0 prior to Phase I.

In Phase 0 we remove vertices from G_w and get a modified instance $\widetilde{G}_w$ that satisfies $\widetilde{c}(x) \geq 0$ for all $x \in \widetilde{V}$. We use a primal-dual 2-approximation algorithm to solve the generalized vertex cover problem on the complete graph $K_{|V|}$ with vertex costs c and edge costs $\overline{w}$. A solution is a set of vertices in the cover, and its cost is the total costs of the vertices in the cover plus the total cost of the uncovered edges. This is clearly a lower bound on the cost of an optimal solution for the clustering instance G_w, because if both elements in a pair e are not outliers, then we need to pay at least $\overline{w}(e)$ in the clustering cost.

We set the vertices in the cover output by the generalized vertex cover algorithm as outliers. The cost of these outliers is at most the cost of the generalized vertex cover solution (which may include also uncovered edges). The total cost of Phase 0 is therefore at most twice the optimum. Now we execute phases I and II on $\widetilde{G}_w$. The combined solution from all phases gives a 17-approximation to the original instance.

1.5 Open Problems

1. Non complete graphs: we can add wight 0, which mean that the edge can be inside a cluster or between clusters without costing us any penalty, this equivalent to work on non complete graphs.

2. Budget: we have a budget B to pay on the outliers, i.e. we are limited on how many vertex we can remove of the graph. The LP will be modified to include the constraint $\sum_{x \in V} p_x \cdot c(x) \leq B$.

Chapter 2

Approximating the Reordering Buffer Management Problem

2.1 Introduction

2.1.1 The Model

The model is an input sequence $\sigma = \sigma_1\sigma_2 \dots \sigma_n$ of objects which are only characterized by specific attribute. To simplify, we suppose that the objects are characterized by their color. The input sequence is processed from left to right by a sorting buffer which is a random access buffer with storage capacity of k objects. During this process, objects may be stored in the buffer and removed later back into the sequence. The resulting sequence is the output sequence. We will denote to the input sequence by the colors of the objects. In the uniform case we will charge a unit for each color change in the result sequence. In the non-uniform case we will charge a cost of b_c for changing to color c. We can state the output objective in two ways. Either we want to maximize the total number of saved color changes or we want to minimize it.

2.1.2 Previous Results

The uniform case is studied in [18, 17]. Räcke, Sohler and Westermann [18], show that several standard strategies, like $FIFO$ and LRU, are unsuitable for a reordering buffer. They later presented a deterministic Bounded Waste strategy that achieves a competitive ratio of $O(\log^2 k)$ in the uniform model. Kohrt and Pruhs [17] present a polynomial-time offline algorithm

that achieves a constant approximation ratio of 20. However, their goal is to maximize the number of saved color changes.

The non-uniform case is studied in [12, 5]. Englert and Westermann [12] present the deterministic MAP strategy and prove that MAP achieves a competitive ratio of $O(\log k)$. The offline maximized variant of the problem was studied by Bar-Yehuda and Laserson [5], they present a polynomial-time offline algorithm that achieves an approximation ratio of 9. Recently, Englert, Röglin and Westermann [11] study the worst case performance of MAP. Based on theoretical considerations and experimental results, they give strong evidence that the competitive ratio of $OPT_{\frac{k}{4}}$ against OPT_k is $\Omega(\sqrt{\log k})$.

2.1.3 Our Contribution

The MAP strategy works as follows. At the beginning of each step it fills the buffer with objects from the input, then it chooses one color as the active color, and continues removing at each step one object of this active color from the reordering buffer until all objects in the buffer have a color different from the active color. Then a new active color has to be chosen. for this purpose a penalty counter P_c, which is initially set to zero, is assigned to each color c. Map chooses as the new active color a color c with $P_c - k \cdot b_c \geq P_{c'} - k \cdot b_{c'}$ for each color c'. The counters are updated after a new active color is chosen. Suppose a step in which a color change from color x to color y occurs. Let n_c denote the number of objects of color c stored in the buffer at the beginning of this step, then each counter P_c is increased by $n_c \cdot b_y$ and counter P_x is reset to zero.

MAP:

1. Initialize $P_c \leftarrow 0$ for all color c, choose $c_{active} \leftarrow c_0$.
2. Fill the buffer with objects from the sequence, as long as there are objects left in the sequence.
3. As long as there are objects of color c_{active} in the buffer, output it.
4. Choose $c_{active} = c$ such that $P_c - k \cdot b_c \geq P_{c'} - k \cdot b_{c'}$ for all color c'.
5. For all color c update $P_c = P_c + n_c \cdot b_{c_{active}}$ and set $P_{c_{active}} = 0$. Go to 2.

The MAP strategy achieves $O(\log k)$ in the non-uniform case of the problem. The proof is divided into two parts:

1. MAP that uses a k-buffer is constant competitive against an optimal offline algorithm that uses a $\frac{k}{4}$-buffer (denote it by $OPT_{\frac{k}{4}}$).

2. $OPT_{\frac{k}{4}}$ costs at most a factor of O(log k) times the cost of OPT_k.

In section 2.2 we show that the 2^{nd} result is tight, i.e. the gap between $OPT_{\frac{k}{4}}$ and OPT_k is $\Omega(\log k)$.

2.2 Proof of the $\Omega(\log k)$ gap

In this section we will show an instance of RBM that can be rearranged using a k-buffer with $N(1+o(1))$ color changes, while it cannot be rearranged using a $\frac{k}{4}$-buffer with less than $O(N \log k)$ color changes.

The instance we use is divided into N interval. Each interval $i \in \{1 \dots N\}$ will contains the objects as follow:

$$\left((c_{i+2^j})^{2^{\alpha-j}} (c_i)^k \right)_{j=0}^{\alpha}$$

where:

- $\alpha = \left\lceil \log(\frac{k}{\log k}) \right\rceil$
- $(c_{i+2^j})^m$: is a sequence of m objects of color c_{i+2^j}. We will call it the j^{th} package of interval i.

The cost of color changing will be 1 unit for all colors, i.e. $b_c = 1$ for all colors c. Thus, our proof holds also for the uniform case.

Lemma 4. *A k-buffer solution can output the objects with $N(1+o(1))$ color changes.*

Proof. It is sufficient to show that for interval i there is place in the k-buffer to hold all the objects with color c_l for $i < l \leq i + 2^\alpha$, that appears in the sequence from interval 1 to interval i (i included).

We will show this by summing these objects according to packages size. e.g. there is just one package of size 2^α (the one of color c_{i+2^0}), and two of

size $2^{\alpha-1}$ (colors c_{i+2^0} and c_{i+2^1}) and so on till we reach 2^α packages of size one. When we sum over all the sizes we get $2^\alpha(\alpha+1)$ objects and this is less than k.

Thus, the OPT_k can output color c_i at interval i while holding in the buffer all colors c_l for $l > i$, that appears in the input until interval i, i.e. OPT_k changes to each color exactly once, and there are $N+\alpha+1$ different colors in the instance. If we choose $N >> \alpha$ then the cost of OPT_k will be $N(1+o(1))$. □

Lemma 5. *The cost of any $\frac{k}{4}$-buffer solution for the above instance is bounded from bellow by $O(N \log k)$.*

Define:

1. x_{i_j} is set to zero iff the j^{th} package of interval i of the input sequence was output together with the $(j-1)^{th}$ package of interval $i+2^j$.
2. x'_{i_j} is set to one iff x_{i_j} is set to zero.
3. $y_{i_j,t}$ is set to one iff the j^{th} package of interval i of the input sequence was in the buffer at interval t. We assume that object is said to be in the buffer at interval t iff it was output after interval t.

Lemma 6. *The following linear program is a lower bound of the cost of any offline strategy that uses $\frac{k}{4}$-buffer on the above instance.*

$$
\begin{array}{lll}
\text{minimize} & \sum_{i=1}^{N}\sum_{j=0}^{\alpha} x_{i_j} & \\
\text{subject to} & -\sum_{i=t-2^\alpha}^{t}\sum_{j=0}^{\alpha} 2^{\alpha-j}\cdot y_{i_j,t} \geq -\frac{k}{4} & \forall t \in \{1..N\}, \quad (2.1)\\
& y_{i_j,t} - x'_{i_j} \geq 0 & \forall i \in \{1..N\}, j \in \{0..\alpha\}, \quad (2.2)\\
& & t \in \{(i+1)..(i+2^{j-1})\},\\
& x_{i_j} + x'_{i_j} \geq 1 & \forall i \in \{1..N\}, j \in \{0..\alpha\}, \quad (2.3)\\
& x, x', y \geq 0. &
\end{array}
$$

Proof. Given a feasible solution for the above instance. Consider the corresponding $\{0,1\}$ assignment to x, x', y according to the above definition. Then constraints (2.1) are satisfied because in each interval t a feasible solution cannot store more than $\frac{k}{4}$ objects in the buffer. Constraints (2.2) are satisfied because if we managed to output the j^{th} package of interval i of the input sequence together with the $(j-1)^{th}$ package of interval $i+2^j$,i.e. $x'_{i_j} = 1$, then the j^{th} package should be stored in the buffer for each interval $t \in \{(i+1)..(i+2^{j-1})\}$,i.e. $y_{i_j,t} = 1$. Constraints (2.3) are satisfied because of the assignment definition. □

And the dual linear program will be:

$$
\begin{array}{lcll}
\text{maximize} & \sum_{i=1}^{N}\sum_{j=0}^{\alpha} w_{i_j} - \frac{k}{4}\sum_{t=1}^{N} v_t & & \\
\text{subject to} & z_{i_j,t} - 2^{\alpha-j}\cdot v_t \leq 0 & \forall i \in \{1..N\}, j \in \{0..\alpha\}, & (2.4) \\
 & & t \in \{1..N\}, & \\
 & -\sum_{t=i+1}^{i+2^{j-1}} z_{i_j,t} + w_{i_j} \leq 0 & \forall i \in \{1..N\}, j \in \{0..\alpha\}, & (2.5) \\
 & w_{i_j} \leq 1 & \forall i \in \{1..N\}, j \in \{0..\alpha\}, & (2.6) \\
 & z, v, w \geq 0. & &
\end{array}
$$

We complete the proof of lemma (5) by showing a feasible dual assignment with a cost of $O(N \log k)$ then we can finish the proof.

Applying constraints (2.5) to constraints (2.4) gives $w_{i_j} \leq \sum_{t=i+1}^{i+2^{j-1}} z_{i_j,t} \leq \sum_{t=i+1}^{i+2^{j-1}} (2^{\alpha-j})\cdot v_t$. That is true for all $i \in \{1..N\}, j \in \{0..\alpha\}$. We will choose an assignment that apply the same value w for all w_{i_j} variables and the same value v for all v_t variables. Thus, we should choose values w, v such that $w \leq \sum_{t=i+1}^{i+2^{j-1}} (2^{\alpha-j})\cdot v = (2^{\alpha-1})\cdot v$. As we need to satisfy constraints (2.6) we will choose $w = 1$ that applies $v = 2^{-(\alpha-1)}$.

It is obvious that the above assignment is a feasible assignment for the dual linear program, it remains to show its cost:

$$
\begin{aligned}
DLP_{\frac{k}{4}} &= \sum_{i=1}^{N}\sum_{j=0}^{\alpha} w_{i_j} - \frac{k}{4}\sum_{t=1}^{N} v_t \\
&= \sum_{i=1}^{N}\sum_{j=0}^{\alpha} 1 - \frac{k}{4}\sum_{t=1}^{N} 2^{-(\alpha-1)} \\
&= N\left(\alpha + 1 - \frac{\log k}{2}\right) \\
&= N\left(\log k - \log(\log k) + 1 - \frac{\log k}{2}\right) \\
&= N \cdot \Theta(\log k)
\end{aligned}
$$

Thus, $\Omega(\log k)OPT_k \leq DLP_{\frac{k}{4}} \leq OPT_{\frac{k}{4}}$ as required.

2.3 Open Problems

1. Give better analysis for MAP or on the contrary give a proof of a non-constant gap.
2. Give online stategy for Reordering buffer that gives better competitive ratio than MAP.

Bibliography

[1] N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: Ranking and clustering. *Journal of the ACM*, pages 684–693, 2005.

[2] N. Alon, S. Dar, M. Parnas, and D. Ron. Testing of clustering. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science*, pages 240–250, 2000.

[3] M. Badoiu, S. Har-Peled, and P. Indyk. Approximate clustering via coresets. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 250–257, 2002.

[4] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Machine Learning Journal (Special Issue on Theoretical Advances in Data Clustering)*, pages 238–247, 2004. A preliminary version appeared in FOCS 2002.

[5] R. Bar-Yehuda and J. Laserson. 9-approximation algorithm for sorting buffers. In *Proceedings of the 3rd Workshop on Approximation and Online Algorithms*, 2005.

[6] M. Charikar, V. Guruswami, and A. Wirth. Clustering with qualitative information. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, pages 524–533, 2003.

[7] M. Charikar, S. Khuller, D. M. Mount, and G. Narasimhan. Algorithms for facility location problems with outliers. In *Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 642–651, 2001.

[8] W. F. de la Vega, M. Karpinski, C. Kenyon, and Y. Rabani. Approximation schemes for clustering problems. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 50–58, 2003.

[9] E. Demaine and N. Immorlica. Correlation clustering with partial information. In *Proceedings of the 6th International Conference on Approximation Algorithms for Combinatorial Optimization Problems*, pages 1–13, 2003.

[10] D. Emanuel and A. Fiat. Correlation clustering — minimizing disagreements on arbitrary weighted graphs. In *Proceedings of the 11th Annual European Symposium on Algorithms*, pages 208–220, 2003.

[11] M. Englert, H. Röglin, and M. Westermann. Evaluation of online strategies for reordering buffers. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, pages 556–564, 2007.

[12] M. Englert and M. Westermann. Reordering buffer management for non-uniform cost models. In *Proceedings of the 32st International Colloquium on Automata, Languages and Programming (ICALP)*, pages 627–638, 2005.

[13] I. Giotis and V. Guruswami. Correlation clustering with a fixed number of clusters. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1167–1176, 2006.

[14] M. X. Goemans and D. P. Williamson. The primal-dual method for approximation algorithms and its application to network design problems. In *Approximation Algorithms, D. Hochbaum, Ed.*, 1997.

[15] R. Hassin and E. Or. Min sum clustering with penalties. In *Proceedings of the 13th Annual European Symposium on Algorithms*, pages 167–178, 2005.

[16] P. Indyk. A sublinear time approximation scheme for clustering in metric spaces. In *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*, pages 154–159, 1999.

[17] J. Kohrt and K. Pruhs. A constant approximation algorithm for sorting buffers. In *Proceedings of the 6th Latin American Symposium on Theoretical Informatics (LATIN)*, pages 193–202, 2004.

[18] H. Räcke, C. Sohler, and M. Westermann. Online sheduling for sorting buffers. In *Proceedings of the 10th Annual European Symposium on Algorithms*, pages 820–832, 2002.

[19] L. J. Schulman. Clustering for edge-cost minimization. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 547–555, 2000.

[20] C. Swamy. Correlation clustering: maximizing agreements via semidefinite programming. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 519–520, 2004.

2.1 עבודות קודמות

אינגלרת, רוגלין וויסטרמאן [11] למדו את פער הקירוב של MAP. בהסתמך על שיקולים תיאורטיים ותוצאות ניסיוניים נתנו הוכחה לכך שהיחס התחרותי של $OPT_{k/4}$ כנגד OPT_k הינו $\Omega\left(\sqrt{\log k}\right)$.

2.2 התרומה שלנו

בתזה, אנו מראים כי התוצאה המוצגת בחלק השני של ההוכחה עבור MAP הינה הדוקה. כלומר, היחס התחרותי של $OPT_{k/4}$ כנגד OPT_k הינו $\Omega(\log k)$.

2. בעיית שינוי סידור בעזרת חוצץ

המודל: רצף של אובייקטים $\sigma = \sigma_1\sigma_2\ldots\sigma_n$ שלכל אחד מיוחסת תכונה מסוימת. מטעמי פשטות, נניח שהתכונה היא צבע. אזי, הרצף בקלט מעובד משמאל לימין בעזרת חוצץ מיון, החוצץ בעל גישה רנדומאלית עם קיבולת של k אובייקטים.

במשך התהליך, אפשר לאחסן אובייקטים בחוצץ ולהוציא אותם יותר מאוחר בחזרה לרצף. הרצף הנוצר הוא הפלט. אנו נסמן את הרצף בקלט לפי הצבע של האובייקטים.

במקרה האחיד, נגבה יחידת תשלום אחת על כל שינוי בצבע ברצף התוצאה. לעומת המקרה הלא-אחיד בו נגבה מחיר של b_c לשינוי לצבע c.

אסטרטגית MAP המוצגת במאמר של אינגלרת וויסטרמאן [12]: מניחה שאובייקטים בקלט יכנסו לחוצץ המיון כל עוד יש מקום ריק בחוצץ. האסטרטגיה בוחרת צבע אחד להיות הצבע הפעיל, וממשיכה צעד אחרי צעד, להוציא אובייקטים עם הצבע הפעיל מחוצץ המיון עד שבחוצץ לא יישאר שום אובייקט מאותו צבע פעיל. ואז צבע אחר נבחר להיות הצבע הפעיל. למטרה זו, מונה עלות P_c, מאותחל לאפס, מוקצה לכל צבע c. MAP בוחרת כצבע הפעיל הבא את הצבע c המקיים: $P_c - k \cdot b_c \geq P_{c'} - k \cdot b_{c'}$ לכל צבע 'c. והמונים מתעדכנים כל פעם שבוחרים צבע פעיל חדש. נסתכל על הצעד בו הצבע הפעיל השתנה מצבע x לצבע y, ונסמן ב- n_c את מספר האובייקטים מצבע c המאוחסנות בחוצץ בתחילת הצעד. אזי, כל מונה P_c גדל ב- $n_c \cdot b_y$ והמונה P_x מאופס.

אסטרטגית MAP משיגה יחס תחרותי $O(\log k)$ במקרה הלא-אחיד של הבעיה. ההוכחה מתחלקת לשני חלקים:

1. MAP משיגה יחס תחרותי קבוע נגד אלגוריתם אופטימאלי לא מקוון שמשתמשת בחוצץ עם קיבול k/4.
2. פתרון אופטימאלי לא מקוון המשתמש בחוצץ עם קיבולת k/4 עולה לכל היותר פקטור $O(\log k)$ פעמים המחיר של פתרון אופטימאלי לא מקוון המשתמש בחוצץ עם קיבול k.

1.2 התרומה שלנו

האלגוריתם שלנו משתמש בסכימה פרימאל-דואל, מבוססת על רלקסציה ליניארית. הרלקסציה שלנו היא עידון של הרלקסציה המוצגת בניתוח הדואלי לאלגוריתם המוצע ע"י אילון-צ'ארקר-ניומאן [1]. הם השתמשו ברלקסציה הבאה:

$$\begin{aligned} \min \quad & \sum_{e \in E_+ \cup E_-} y_e \\ \textit{subject to} \quad & \sum_{e \subset t} y_e \geq 1 \qquad \forall t \in T \\ & y_e \geq 0 \qquad \forall e \in E_+ \cup E_- \end{aligned}$$

כאשר T מסמן את קבוצת המשולשים ב-G. כלומר, קבוצת השלישיות $\{x_1, x_2, x_3\}$ שמכילה שני יחסים מסוג "+" ויחס אחד מסוג "-".

הנקודה המעניינת באלגוריתם שלנו היא השימוש בסכימת הפרימאל-דואל רק למטרה אחת, והיא לזיהוי קבוצת הצמתים שיוצאו מהסיווג תוך תשלום המחיר שלהם. בעצם, השלב הזה של האלגוריתם אינו מוציא פתרון פיזיבילי לסיווג. (לרלקסציה הליניארית יש פתרונות פיזיביליים בשלימים שאינם קשורים לאף פתרון פיזיבילי עבור בעיית הסיווג עם מחירים; זה גם נכון עבור הרלקסציה ב-[1]). יש להדגיש כי זיהוי קבוצת הצמתים הכי טובה להסרה מהסיווג הינה בעיה NP-קשה.

לאחר הורדת קבוצת הצמתים שמצאנו בשלב הראשון. בשלב השני מריצים את אלגוריתם אילון-צ'ארקר-ניומאן שמשיג קירוב 3 על הצמתים הנותרים. ומקבלים את הסיווג של הפלט. המחיר הכולל של מחיר הסיווג בתוספת מחיר החורגים הוא קירוב 9 לבעיית הסיווגים עם מחיר על הצמתים.

בנוסף אנו מראים עדכון לאלגוריתם שנותן פתרון עם קירוב 17 לבעיית הסיווג עם מחירים על הצמתים ומשקולות על הקשתות.

"+" או "-", שמסמן את הרצון של הקשת שהצמתים שלה יהיו באותה קבוצת סיווג או בשתי קבוצות סיווג שונות, בהתאם. המטרה היא חלוקה של קבוצת הצמתים לקבוצות סיווג שמתחשבת בסימונים האלה כמידת האפשר. אחת האטרקציות בניסוח של בעיית הסיווג הזו היא שלא נדרש לקבל את מספר קבוצות הסיווג k מראש. אלא שהמספר הזה מושרה מפונקצית העלות.

ברור שאפשר להגדיר את המטרה של הסיווג בשתי צורות, או שנרצה למזער את המספר הכולל של קשתות המסומנות הפוך לסיווג (כלומר, קשתות המסומנות "+" עם קצבות בקבוצות סיווג שונות וקשתות המסומנות "-" עם קצבות באותה קבוצת סיווג). או שנרצה למקסם את המספר הכולל של קשתות המסומנות בצורה עקבית עם הסיווג.

המחיר של פונקציות המטרה שווה באופטימום, הבעיה הינה NP קשה. הם עלולים להיות (והם אכן) שונים כל עוד נדרש קירוב. בגלל שאנו מעוניינים במודל עם חורגים, בתזה זו נתרכז בגרסת המזעור של הבעיה.

1.1 עבודות קודמות

רצף של מאמרים נתן אלגוריתמי קירוב לשתי הגרסאות של הבעיה. (כמו כן, להכללות של הבעיה, איפה שרק תת קבוצה של הקשתות הינן מסומנות) [4,6,9,10,20,1,13]. הקירוב הקבוע הראשון לגרסת המזעור בגרפים מלאים ניתן ע"י בנסל, בלאם, וצ'אולה [4]. הפקטור הזה שופר ל-4 בעזרת טכניקת העיגול לתוכנית ליניארית ע"י צ'אריקר-גורוסואמי-וירת' [6]. התוצאה הטובה ביותר לגרסת המזעור נותן האלגוריתם הרנדומאלי החמדן המשיג קירוב 3 שהוצג ע"י אילון-צ'ארקר-ניומאן [1].

$CC-Pivot\left(G=\left(V,E_{+}\right)\right)$:

- בחר צומת אקראי $v\in V$.
- תאתחל $C=\left\{v\right\},V'=\varnothing$.
- לכל צומת $u\in V$ עבור $u\neq v$:
  - אם $\left(v,u\right)\in E_{+}$
    - תוסיף את u ל-C.
  - אחרת ($\left(v,u\right)\in E_{-}$)
    - תוסיף את u ל-V'.
- נסמן ב-G' את תת הגרף המושרה ע"י V'.
- תחזיר את הסיווג $C,CC-Pivot\left(G'\right)$

E_{+} היא קבוצת הקשתות עם היחס "+". E_{-} היא קבוצת הקשתות עם היחס "-".

תקציר

1. בעיית סיווג בגרפים עם משקולות

ניסוחים שונים של בעיות הסיווג תמיד היו רגישים לקיום החורגים, שמהווים נקודות רעש אקראיים המקלקלים את הפתרון הנקי לחלוקת קבוצות הנתונים.
חלק מאלגוריתמי הקירוב לסיווג נתונים במרחבים עם מימד גבוה, תחת פונקציות מטרה מגוונות, אפשר להתאים אותם על מנת לטפל בחורגים בצורה טובה. לרוב על ידי כך שמתירים הסרת חלקי נקודות (ראה, כדוגמה [19,2,3,8]). עם זאת, תוצאות כאלה נחשבות נדירות לבעיות סיווג בגרפים.
בתזה הזו, אנו לומדים מדד סיווג שמאפשר קיום חורגים. המדד הוא מודל איסוף עם מחירים, שנלמד מוקדם יותר בהקשר של בעיית תכנון רשתות (ראה [14]).

קבוצת הנתונים מיוחסת עם פונקצית מחיר, הפלט המסווג יכול לכסות כל תת קבוצה של קבוצת הנתונים, ומטרת הסיווג היא למזער את הסכום של פונקצית המטרה הסטנדרטית על הנתונים המסווגים בתוספת המחיר הכולל של הנתונים הלא מסווגים.

מחיר של צומת יכול לייצג את דרגת הביטחון בכל נקודת נתונים. ביטחון גבוה מתפרש במחיר גבוה על אי שיתוף הנקודה בסיווג.

מקודם, צ'אריקר ועמיתיו [7] בחנו את בעיית "מיקום מפעלים עם חורגים". הם הציגו אלגוריתם שנותן קירוב 3 לבעיית מיקום המפעלים עם מחירים, ואלגוריתם שנותן קירוב 4 לבעיית "k-חציון עם מחירים". חאסין ואור [15] למדו לאחרונה את התחום הזה בהקשר של סיווג ע"י מזעור עלות קשת: בהינתן גרף מלא $G=(V,E)$, פונקצית משקל על הקשתות $w:E\to N$, ופונקצית מחיר על הצמתים $c:V\to N$, המטרה היא לחלק את V ל- k קבוצות סיווג, $\{S_1,...,S_k\}$ וקבוצת חורגים U, כך שממזערים את סכום המשקולות של הקשתות במולטי-חתך (החותכות את קבוצות הסיווג) ועוד המחיר של החוגרים. הם נתנו אלגוריתם קירוב 2 למקרה של סיווג לקבוצה אחת. נתנו PTAS לסיווג יחיד של מטריקה עם מחירים אחידים. וקירוב 2 של מטריקה עם מספר קבוע של קבוצות סיווג. (אם בנוסף איננו מתירים חוגרים, אז יש קירוב PTAS של מטריקה עם מספר קבוע של קבוצות סיווג [8,16]).

אנחנו חוקרים מדד איסוף עם מחירים בהקשר של סיווג מתאם, ניסוח תיאורטי בגרפים שנלמד לראשונה ע"י בנסל, בלאם, וצ'אולה [4]. הבעיה המקורית מוגדרת באופן הבא, יש לנו גרף מלא שבו הקשתות מסומנות ב-

המחקר נעשה בהנחיית פרופ'/ח' יובל רבני מהפקולטה למדעי המחשב.

ברצוני להודות תודה עמוקה לפרופ'/ח יובל רבני על הנחיתו, תמיכתו, עזרתו ועל שדחף אותי קדימה בעת הצורך.

בנוסף, ברצוני להודות למשפחה שלי על האהבה והתמיכה שנתנה לי, ולחברים שלי על כל הזמנים הטובים שהיו לנו.

אני מודה לטכניון על התמיכה הכספית הנדיבה בהשתלמותי

בעיית סיווג בגרפים עם משקולות
ובעיית שינוי סידור בעזרת חוצץ

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר

מגיסטר למדעים במדעי המחשב

אמג'ד עבוד

הוגש לסנט הטכניון - מכון טכנולוגי לישראל

שבט תשס"ח חיפה ינואר 2008

בעיית סיווג בגרפים עם משקולות
ובעיית שינוי סידור בעזרת חוצץ

אמג'ד עבוד