Expected Approximation Guarantees for the Demand Matching Problem

C. Boucher D. Loker

September 2006

Abstract

The objective of the demand matching problem is to obtain the subset M of edges which is feasible and where the sum of the profits of each of the edges is maximized. The set M is feasible if for each vertex v the total demand of edges in M incident to v is at most b_v. In the case where each of the edges has one unit profit, the problem becomes finding the subset of largest size and hence, is called the cardinality problem. Shepherd and Vetta [SV06] demonstrate that the integrality gap for the general demand matching problem for bipartite graphs is between 2.5 and 2.764, and between 3 and 3.264 non-bipartite graphs. We demonstrate that an expected 2.5-approximation guarantee and 3-approximation guarantee is achieveable for bipartite graphs and non-bipartite graphs and give some connections to the independent set and weighted independent set problem.

1 Introduction

An instance of the *demand matching problem* is defined as an undirected graph $G = (V, E)$ where each edge $e \in E$ has an associated *profit* p_e and an associated demand d_e, and each vertex $v \in V$ has an associated *bound*, b_v. A set of edges M is feasible if for each vertex v the total demand of the edges in M incident to v does not exceed the bound associated with v. The aim of the demand matching problem is to find the optimal feasible set M^*, that is, the set of edges that are feasible and have maximum profit. The problem is a specialization of the b-matching problem on a graph; an additional constraint is needed that ensures each edge has an associated demand value. This

NP-hard combinatorial maximum packing problem is well-known and there are vast number of applications to this theoretical problem; a short list of application include the design of network switches [SV06], resource allocation [CCHR02], and requests scheduling [LMV00].

A traditional method to solve packing problems, such as the demand matching and b-matching problems, consists of considering a linear programming formulation of the form $\{p \cdot x : Ax \leq b, x \geq 0\}$ for the given problem and attempting to find a $0-1$ vector indicating an optimal or near-optimal solution. The demand matching problem can be formulated as the optimization problem: max$\{p \cdot x : Dx \leq b, x_i \in \{0,1\}\}$, where p is the profit vector, D indicates the associated demand on the edges, and x is a $0-1$ indicator vector for the edges. We will restrict interest to the "all or nothing" version of the demand matching problem, implying that we require that the whole demand of the edge must be satisfied to gain the profit or none of the demand, and thus profit, of the edge is taken. Shepherd and Vetta note that analysis of the linear relaxations for this demand matching integer programming formulation require different methods from those used for the b-matching problem [SV06]. For a more comprehensive study of the connection between the demand matching problem and other combinatorial maximum packing problems, we refere the reader to the work by Shepherd and Vetta [SV06].

Study of this particular packing problem has been rather limited. A randomized algorithm exists which provides a 2.764-approximation guarantee for the demand matching problem on bipartite graphs, and a separate randomized algorithm for non-bipartite graphs provides a 3.264-approximation guarantee [SV06]. If interest is restricted to the demand matching problem where the profit on each edge is 1, a greedy algorithm provides a 2.5-approximation guarantee for bipartite graphs, and a randomized algorithm exists which provides a 3-approximation guarantee for non-bipartite graphs [SV06]. Currently, these are the best known approximation guarantees for both the demand-matching problem. Further, Vetta and Shepherd [SV06] demonstrate that the demand matching problem, and the cardinality version of the problem, is MAXSNP-hard; implying that there exists a constant $\epsilon > 0$ such that the problem admits no $1 + \epsilon$-approximation algorithm, unless P = NP. The focus of this paper will be improving upon the existing approximation guarantees for the general demand matching problem by showing there exists an algorithm with expected 3-approximation guarantee for non-bipartite graphs and an algorithm with 2.5-approximation guarantee for non-bipartite graphs.

An *independent set* of a graph G is a subset of the vertices such that no two vertices in the subset represent an edge in G. A maximum independent set is a largest independent set for a given graph. The problem of finding such a set is called the MAXIMUM INDEPENDENT SET (MIS) problem and is an NP-hard problem [GJ79], whose approximability has been intensely investigated [BH90, Has99, Hal00]. A maximum independent set should not be confused with a *maximal independent set*, which is an independent set that is not contained in any larger independent set. The problem of finding a maximal independent set can be solved in polynomial time by a trivial greedy algorithm. WEIGHTED INDEPENDENT SET (WIS) is the problem of finding the independent set of maximum weight, this implicitly assumes that each vertex has a non-negative weight. Our approximation bound for the general demand matching problem relies on a transformation from the demand matching problem to an equivalent weighted independent set problem. We present some approximation results for WIS and MIS and demonstrate their connection to the demand matching problem. The main focus of this paper will be an expected 2.5-approximation guarantee and an expected 3-approximation guarantee for bipartite graphs and non-bipartite graphs, respectively.

2 The Demand Matching Problem

Let $G = (V, E)$ be a graph and assign each vertex $v \in V$ an integral *capacity* denoted as b_v, and each edge $e = (u, v) \in E$ an integral demand d_e (for any edge e adjacent to a vertex v we assume that $d_e \leq b_v$, otherwise we simply remove e). In addition, associated with each edge $e \in E$ is it's *profit* p_e. A demand matching is a subset $M \subseteq E$ such that $\sum_{e \in \Gamma(v) \cap M} d_e \leq b_v$ for each vertex v, where $\Gamma(v)$ denotes the neighbours of v. The demand matching problem is to find a demand matching of maximum profit and thus, it can be formulated as the following integer program (IP):

$$\begin{aligned} \text{maximize:} \quad & \textstyle\sum_{e \in E} p_e x_e \\ \text{subject to:} \quad & \textstyle\sum_{e \in \Gamma(v)} d_e x_e \leq b_v, \quad \forall v \in V \\ & x_e \in \{0, 1\} \qquad\qquad \forall e \in E \end{aligned}$$

An alternative formulation of the demand matching problem is associated with the *marginal profit* of an edge, which is $\pi_e = p_e/d_e$. The following is an alternative IP formulation that we will refer to as the *marginal profit IP*.

$$\begin{aligned} \text{maximize:} \quad & \textstyle\sum_{e \in E} \pi_e x_e \\ \text{subject to:} \quad & \textstyle\sum_{e \in \Gamma(v)} x_e \leq b_v, \quad \forall v \in V \\ & x_e \in \{0, d_e\} \qquad \forall e \in E \end{aligned}$$

The linear program (LP) relaxation of the marginal profit IP replaces the integral constraint on the x_e values by the linear constraint $0 \leq x_e \leq d_e$. The solution space of the resultant LP is the *fractional demand matching polytope* and hence, a point $\mathbf{x}$ in the polytope is a *fractional demand matching*. For a fractional demand matching $\mathbf{x}$ a vertex v is referred to as *tight* if $\sum_{e \in \Gamma(v)} d_e x_e = b_v$ and otherwise, the vertex is referred to as *fractional*. Similarly, we refer to the an edge as *tight* if $x_e = d_e$ and otherwise we have $0 < x_e < d_e$ and e is referred to as *fractional*. We let $F(\mathbf{x}) \subseteq E$ be the set of fractional edges induced by the fractional demand matching $\mathbf{x}$ and let $G(\mathbf{x})$ denote the graph induced by $F(\mathbf{x})$. The following lemma demonstrates that there is a significant amount of structure in the LP solution and hence, shows that $G(\mathbf{x})$ for bipartite graphs exhibits a tree structure.

Lemma 1 *[SV06] Let $\mathbf{x}$ be an extreme point of the demand matching polytope. Then each component of $G(\mathbf{x})$ consists of a tree plus (possibly) one edge. In addition, any cycle in $G(\mathbf{x})$ has odd length.*

2.1 The Cardinality Problem

When interest is restricted to the demand problem where the demand on every edge is equal to one, the problem becomes solvable in polynomial-time. However, the problem remains NP-hard when interest is restricted to the demand matching problem where there exists unit profit on every edge. The demand matching problem which has this unit-profit restriction is known as the *cardinality problem*. Previous results demonstrate that improved approximation guarantees may exist for algorithms for the cardinality problem, in comparison to the traditional demand matching problem. The best approximation guarantee for the cardinality problem for bipartite graphs is a factor 2.5-approximation guarantee and for non-bipartite a factor 3-approximation guarantee [SV06].

2.2 Previous Approximation Guarantees and Hardness Results

Calinescu *et al.* [CCHR02] proposed the open problem of determining whether demand matching is MAXSNP-hard or whether there exists a polynomial-time approximation scheme. Recently this open problem has been solved by Shepherd and Vetta [SV06]: as they demonstrate that the cardinality problem–and hence, the general demand matching problem–is MAXSNP-complete. Hence, there does not exist a constant $\epsilon > 0$ such that the problem admits no $1 + \epsilon$-approximation, unless P = NP.

There has been limited study on the demand matching problem and more generally, on demand versions of combinatorial packing problems. A special case of the generalized assignment problem where each task has a size independent of its assignment is a special case of the demand matching problem that has been well-studied. The base problem of this assignment problem can be viewed as a bipartite b-matching problem with the following property: each vertex v of one bipartition has a common demand value on each of its incident edges and its bound b_v is equal to this demand value. The generalized assignment problem was studied by Shmoys and Tardos [ST93], where they studied congestion minimization. The maximization form of this problem was later studied

Chekuri and Khanna [CK00], where these previous minimization results were considered. The techniques and results from Shmoys and Tardos [ST93] and Chekuri and Khanna [CK00] resemble the latest results on the demand matching problem by Shepherd and Vetta [SV06].

Cosares and Caniee [CS94], and Kleinberg [Kl96] study demand versions of network flows. Kleinberg [Kl96] introduces *unsplittable flow* and studies the maximization forms the demand version network flows. In particular, Kleinberg [Kl96] studies the maximization single-source unsplittable flow problem, where a single source s is given, along with a collection of terminals $t_1, t_2, \ldots, t_k$ with demands $d_1, d_2, \ldots, d_k$. The goal of the packing problem is to satisfy the maximum number of the demands subject tot he edge capacity constraints. This unsplittable flow problem can be considered as a demand framework as follows: let each s to t_i path have demand d_i, and add a sink vertex t and edges $t_i t$ with capacity d_i then the goal is to find a maximum packing of the weighted s to t paths. Further, the demand matching problem has connections to other unsplittable flow problems. Kolliopoulos and Stein [KS02] demonstrate the first approximation algorithm for maximum unsplittable network flow problem with capacities. Recently, this work was extended to give a constant factor approximation for the maximum profit unsplittable flow problem where the underlying graph is a tree [CMS03]. Shepherd and Vetta [SV06] note that the maximum unsplittable flow problem, in the case where the tree is a star, includes and the demand matching problem.

As mentioned previously, there is a close connection between the demand matching problem and other more applied problems; one such example is the design of communication switches where the goal is schedule transmission of traffic using a minimum number of time slots. A communication switch can be modelled as a bipartite graph, with the bipartition of the vertices being the input and output ports. With this graph-theoretic model, the goal becomes colouring the edges so that each colour class is a demand matching. For a more comprehensive discussion on this topic we refere the reader to the work by Ngo and Vu [NV03].

The best known approximation guarantees for the general demand matching problem are a

2.764-approximation and a 3.264-approximation for bipartite and non-bipartite graphs, respectively [SV06]. Shepherd and Vetta [SV06] also give improved approximation guarantees for the cardinality problem: a 2.5-approximation and 3-approximation algorithm for bipartite and non-bipartite graphs, respectively. The results for the general demand matching problem arise from a randomized approximation algorithm, whereas the results for the cardinality problem arise from a deterministic algorithm. The lower bound on the integrality gap is at least 3 for non-bipartite graphs and 2.5 for bipartite graphs [SV06]. These bounds on the integrality gap demonstrate that the previous best known approximation guarantees of 2.5 and 3 for bipartite and non-bipartite graphs, respectively, for the cardinality problem are optimal approximations amongst algorithms that use the LP solution. Although, an improved approximation algorithm for the cardinality problem may exist, it cannot use only the LP solution as the lower bound in estimating the approximation ratio.

3 An Expected Approximation Guarantee

We present an algorithm that gives an expected 2.5-approximation guarantee for bipartite graphs, and an expected 3-approximation guarantee for non-bipartite graphs. As stated previously, the best known approximation guarantee for the demand matching problem was 2.764 for bipartite graphs and 3.264 for non-bipartite graphs [SV06]. Our algorithm works on an alternate formulation of the problem–that is, an equivalent independent set problem which we define in the following sections.

3.1 Applying Berge's Augmenting Paths Conditions to Demand Matching

In 1957, Berge showed a simple characterization of a matching of maximal cardinality which states that a matching is of maximum cardinality if and only if it has no augmenting path [Ber57]. An equivalent result for demand matching is shown by Shepherd and Vetta [SV06] which gives optimality conditions for the fractional demand matching problem. We consider a partition of E into *positive edges*, denoted as $\mathcal{P}$, and *negative edges*, denoted as $\mathcal{N}$ and define the marginal value of a set $F \subseteq \mathcal{P} \cup \mathcal{N}$ as $\pi(F) = \sum_{e \in F \cap \mathcal{P}} \pi_e - \sum_{e \in F \cap \mathcal{N}} \pi_e$.

Definition 1 *We state that there exists an* augmenting path P *in the fractional demand matching if there is a partition* $\mathcal{P} \cup \mathcal{N}$ *of* P *such that (i) the edges on the path alternate between being positive and negative, (ii)* $\pi(P) > 0$*, (iii) if an endpoint* v *of the path is tight then the edge which* v *is adjacent to is negative, and (iv) none of the positive edges in* P *are tight.*

The conditions for an augmenting path in a demand matching ensure that we can improve the current demand matching by *augmenting* around the structure–that is, add ϵ to the positive edges and subtract ϵ from the negative edges. We then have a characterization of an optimal fractional demand matching: a fractional demand matching is optimal if and only if it induces no augmenting structure [SV06].

3.2 Colouring the Edges According to the LP Solution

We will describe a procedure that takes a fractional demand matching and generates a 2-colouring of the edges for bipartite graphs and a 3-colouring for non-bipartite graphs. This procedure will be used in proving the improved bound for the general demand matching problem.

We consider a LP solution, $\mathbf{x}$, for the fractional demand matching problem on a bipartite graph. We will demonstrate a procedure that, for a tree, will find two integral demand matchings whose combined profit is at least that of the original solution; this will in turn give an integral solution whose profit is at least half of the optimal fractional demand matching $\mathbf{x}$ of the tree. Recall that $F(\mathbf{x})$ consists of a set of trees. We denote $\mathbf{x} = \mathbf{f} + \mathbf{h}$, where $\mathbf{f}$ is the set of fractional edges in $\mathbf{x}$ not equal to 0 and $\mathbf{h}$ is the set of tight edges of $\mathbf{x}$. We note that $\mathbf{f}$ is a feasible fractional demand matching and therefore, there exists an augmenting path in the graph induced on the set $\mathbf{f}$. In order, to create an improved demand matching, we augment along an augmenting path in $\mathbf{f}$ such that at least one edge becomes tight or becomes zero. We repeat the process of augmenting along fractional paths until the set of fractional edges induce a demand matching. We call this process of modifying the LP the *augmenting paths procedure* and note it is guaranteed to terminate since at each step we discard an edge or make an edge tight.

The profit of the solution obtained after the augmenting paths procedure is greater than or equal to the profit of the original fractional solution since the profit of the solution is improved at each iteration. Moreover, we note that the set of edges made tight from the set **f** for the augmenting paths procedure, together with the set **h**, do not necessarily induce a feasible demand matching but this does not violate our aim of obtaining two disjoint demand matchings from this union when our underlying graph is a tree. Edges of the final solution are divided into two sets: F^* which is the set of fractional edges, and H^* which is the set of tight edges.

We now colour the edges according to the grouping F^* and H^*; an edge e is *bronze* with respect to an incident vertex v if either e is in F^* or v is not incident to an edge in F^* and e was the last edge to become tight in the augmenting paths procedure. An edge is bronze if it is bronze wither respect to either of its incident vertices and otherwise, it is coloured *copper*.

Lemma 2 *[SV06] The set of copper edges with respect to a vertex v are feasible with regards to the associated vertex constraint.*

Using Lemma 2 it can trivially shown that given a tree T, T contains two disjoint demand matchings M_1 and M_2 whose combined profit is at least that of the optimal fractional demand matching [SV02]. Hence, there trivially exists a 2-approximation algorithm when the underlying graph is a tree.

We now extend the edge-colouring procedure to non-bipartite graphs. For non-bipartite graphs, it follows from Lemma 1 that each component of the optimal fractional demand matching is a tree with possibly one edge creating an odd cycle. As before, we colour tight edges copper and augment along the fractional paths such that the profit of the fractional demand matching is improved at each it step. In addition, we add the possibility of augmenting along an odd cycle. Further, if some edge e on the odd cycle has the property that $x_e \leq \frac{1}{2}d_e$ then we colour e *red* and remove it from the component on an iteration. If no edge has this property then there must exist an edge e with the property that $d_e - x_e \leq \min(x_{e_1}, x_{e_2})$ where e_1 and e_2 are the incident edges to e in the cycle, we colour e copper and remove it from the component. The remainder of the augmenting paths procedure remains as described previously and hence, we guarantee that the algorithm terminates with each of the fractional edges coloured copper, bronze or red.

Hence, Lemma 2 implies that the set of copper edges induce a feasible–but not necessarily optimal–solution to the demand matching problem. Finding an optimal approximation solution to the demand matching problem is then narrowed to finding a set of (bronze) edges that can be made tight and added to the set of copper edges, to obtain a demand matching which has adequate profit. Using Lemma 2 and Lemma 1 a 3-approximation algorithm for bipartite graphs and 3.5-approximation algorithm for non-bipartite graphs for the general demand matching problem can be obtained trivially.

3.3 Transforming Demand Matching to an Independent Set Problem

We now consider transforming the demand matching problem to an equivalent independent set problem which simplifies finding an appropriate randomized approximation algorithm. We will first consider the case in which the original graph is bipartite and thus, each of the edges is coloured copper or bronze. We can easily extend our transformation to the non-bipartite case where we are required to consider the red edges. The independent set problem will be defined on a subgraph of the line graph of G, which we will denote as G'. The vertices of G' are coloured bronze or copper according to the bronze and copper edge colouring of G. Hence, the existence and colouring of the vertices and edges in G' depend on the bronze and copper edge colouring of G. An edge $e = uv$ exists in G' if and only if the edges associated with u and v, denoted as e_u and e_v respectively, share an endpoint v', and at least one of e_u and e_v is bronze with respect to v'. It follows from Lemma 2 that a independent set in G' corresponds to a demand matching in G. We further note two important facts: the bronze vertices in G' induce a forest, and each copper vertex has degree at most 2 in G'. The latter fact follows from the point that for a copper edge uv it may not be possible to place that edge in a demand matching with either of the edges that are bronze with respect to u and v. It follows that we aim to find the independent set of largest profit. For the remainder of this paper we will consider algorithms for the described independent set problem and consider the results in the context of the original demand matching problem.

3.4 An Approximation Algorithm for Uniform Random Instances

In this section, we present an approximation algorithm for the demand matching problem on uniform random instances. Specifically, we give an algorithm that an expected 2.5-approximation for bipartite graphs and an expected 3-approximation guarantee for non-bipartite graphs. We consider a random instance of the general demand matching problem, find the LP solution, and consider it's equivalent independent set problem. As previously stated, each copper vertex c in G' is adjacent to at most two vertices, which we denote as u and v. A subtle consequence of the definition of the independent set transformation in Section 3.2, is that odd cycles can occur in G' and therefore, when u and v are from the same bronze tree, the path between them in their tree has either even or odd length.

We refer to an *uniform random instance* as an instance of the general demand matching problem chosen uniformly from all the set of all feasible instances to the corresponding LP defined in Section 1. Hence, we assume that there is a uniform distribution over all possible $|E'|^2$ independent set problems transformed from the original demand matching problem. This model of the distribution of the independent set problem instances derived from the demand matching problem instances is valid since the most commonly studied model, the $G(n, p)$ random graph model, assumes each possible edge exists independently with probability p.

A given copper vertex c in a given random uniform instance; c is possibly adjacent to a single bronze vertex, no bronze vertices, or two bronze vertices. We consider the case where c is adjacent to two bronze vertices, which we denote as u and v. As previously stated, the path between u and

v maybe odd or even. To analyze the expected approximation ratio of the algorithm defined in this section we will divide all demand matching instances into *bad instances* and *good instances*, the distinction between good and bad instances is based on the probability that the path between u and v is odd. We define the probability that the path between u and v is odd as $\hat{p}$. We define an instance to be good if $\hat{p}$ is greater than or equal to $\frac{1}{2}$ and otherwise we refer to the instance as bad.

Previously the best known approximation guarantee for the demand matching problem was 2.764 [SV06]. The below theorem demonstrates that modifying the algorithm slightly, an expected 2.5-approximation guarantee is achieveable for bipartite graphs.

Theorem 1 *There exists a polynomial-time algorithm, which when applied to uniform random instances of bipartite graphs has an expected approximation ratio of at most 2.5.*

Proof. Given an instance of demand matching problem, we consider the equivalent independent set problem formed from definition in Section 3.2. We select an independent set among the bronze vertices by considering each tree T in the forest of bronze vertices separately and selecting a bipartition at random of T and adding the bronze vertices of the corresponding independent set. We then include all copper vertices such that neither of their bronze neighbours are in in the chosen independent set. The result is clearly a independent set and hence, we have only to analyze the performance of the algorithm.

We calculate the probability that a given vertex is chosen as part of the independent set and therefore, the probability that the corresponding edge is in the demand matching. If the profit obtained from the copper vertices alone, denoted $p(C)$, corresponds to at least $\frac{2}{5}$ of the total profit of the optimal linear program solution, we are done. Similarly, if the profit obtained from the bronze vertices, denoted $p(B)$, corresponds to at least $\frac{4}{5}$ of the total profit of the optimal LP solution then we are done [SV06]. Therefore, it follows that we can assume that $\frac{3}{5} \cdot OPT < p(B) < \frac{4}{5} \cdot OPT$ and $\frac{1}{5} \cdot OPT < p(C) < \frac{2}{5} \cdot OPT$.

For any bronze vertex v we include its bipartition class with probability $\frac{1}{2}$ and hence, v is included in the independent set with probability $\frac{1}{2}$. We now consider the copper vertices. If a copper vertex c is not adjacent to any bronze vertices then c is added to the existing independent set. If c is adjacent to a single bronze vertex then c is included in the independent set with probability $\frac{1}{2}$, since that is the probability that the bronze vertex was not included in the independent set. Otherwise, c is adjacent to two bronze vertices, which we denote as u and v. If u and v are not in the same bronze tree then the probability of each one being added to the independent set is independent of one another and hence, the probability of c being included is $\frac{1}{4}$. Otherwise, u and v are connected by a path within the same tree T and we denote $\hat{p}$ as the probability that this path is of odd length.

If the path is of odd length then u and v must be in opposing bipartition classes and hence one or the other must always be included in the independent set and it follows, that c cannot be included in the independent set. If the path is of even length then u and v are in the same bipartition class and hence, are included in the independent set with probability $\frac{1}{2}$ and therefore, the probability that c is added is $\frac{1}{2}$. The following defines the expected profit from the set of copper vertices:

$$
\begin{aligned}
E[\text{ profit of copper vertices }] &= \min\{\frac{1}{4} \cdot p(C), \hat{p} \cdot 0 + (1-\hat{p}) \cdot \frac{1}{2} \cdot p(C)\} \\
&\geq \min\{\frac{1}{4} \cdot p(C), \frac{1}{2} \cdot 0 + \frac{1}{4} \cdot p(C)\} \\
&= \frac{1}{4} \cdot p(C)
\end{aligned}
$$

From the above analysis, the expected profit we obtain is at least $p(C) \cdot \frac{1}{4} + p(B) \cdot \frac{1}{2}$. This is a continuous function defined as $E \geq x \cdot OPT \cdot \frac{1}{4} + (1-x) \cdot OPT \cdot \frac{1}{2}$, where $x \in (\frac{1}{5}, \frac{2}{5})$. The expected value can be simplified to $E \geq -\frac{1}{4} \cdot OPT \cdot x + \frac{1}{2} \cdot OPT$. The possible range of values then are $(\frac{2}{5} \cdot OPT, \frac{9}{20} \cdot OPT)$, and it follows then that $E \geq \frac{2}{5} \cdot OPT$. □

We now directly apply our previous result to obtain an expected 3-approximation guarantee for non-bipartite graphs. Previously the best known approximation guarantee for non-bipartite graphs was 3.264 [SV06].

Theorem 2 *There exists a polynomial-time algorithm, which when applied to uniform random instances of non-bipartite graphs has an expected approximation ratio of at most 3.*

Proof. We recall that the red edges have the property that $x_e \leq \frac{1}{2}d_e$. If the profit of the red edges constitutes at least $\frac{1}{3}$ the total profit of the LP solution then we are done proving our claim. Thus, we assume that profit induced by the bronze and copper edges is at least $\frac{5}{6} \cdot OPT$. Considering the corresponding independent set problem induced on the edges of G, we omit all red vertices and apply the previous randomized algorithm to the modified independent set problem. The application of the previous algorithm produces a solution whose expected value is $\frac{1}{3} \cdot OPT$. □

4 Connection to Independent Set Problems

NC is the class of problems for which there exists a parallel algorithm that uses $\log^{O(1)} n$ phases of $n^{O(1)}$ simultaneous parallel operations. Further, RNC is the class of problems for which there exists parallel a Monte Carlo algorithm with one-sided error that uses $\log^{O(1)} n$ phases of $n^{O(1)}$ simultaneous parallel operations [MR95]. Luby shows that MIS is in RNC by giving a Monte Carlo algorithm and in NC by converting the Monte Carlo algorithm into a simple deterministic algorithm with the same running time [Lub86]. Using this result by Luby, it follows that approximating the cardinality problem is in RNC and NC. Further, the algorithm by Luby [Lub86] for MIS can be used to obtain an $2\frac{1}{2}$-approximation and 3-approximation for the cardinality problem for bipartite graphs and non-bipartite graphs, respectively.

There have been a number of parallel approximation algorithms and heuristics developed for WIS and investigations into bounding the inapproximability [BH90, Hal00]. In Section 3.2 we developed a method to transform the general demand matching problem into an equivalent independent set problem and hence, the goal of the transformed problem is to find an independent set in G' with largest profit. We now give several graph theoretic definitions and results, and thus, demonstrate how approximation of these results for WIS can be applied to the demand matching problem.

A graph is δ- *inductive* if there is a linear ordering of the vertices such that each vertex has at most δ neighbours ordered after itself [Hal00]. For example, the equivalent independent set graph G' transformed from an optimal fractional demand matching solution defined on a bipartite graph is a 2-inductive graph, since the vertices can be ordered as follows: copper vertices, leaves of the bronze tree, bronze vertices that are adjacentnt to the leaves, $\ldots$, root of the bronze tree. The Lovász number $\varepsilon(G)$ of a graph G is the least number k such that there exists a representation of unit vectors v_i to each $i \in V$, such that for any two nonadjacent vertices i and j the dot product of their vectors is equal to $-\frac{1}{k}$. Given a graph G, $\varepsilon(G)$ can be computed in polynomial time [Hal00]. Halldórsson [Hal00] proves the following theorem of weighted δ-inductive graphs.

Theorem 3 *[Hal00] Let G be a weighted δ-inductive graph satisfying $\varepsilon(G) \leq k$. Then an independent set in G of weight $\Omega(w(G)/\delta^{1-\frac{1}{2k}})$ can be constructed with high probability in polynomial time, where $w(G)$ denotes the sum of the weights of the vertices.*

Thus, by computing $\varepsilon(G)$ for the 2-inductive graph G' we can use Theorem 3 to construct an independent set of G' with weight greater than or equal to $w(G)/2^{1-\frac{1}{2k}}$. Hence, this demonstrates how approximation results for WIS can be applied to yield an approximation to the demand matching problem.

5 Conclusions and Open Problems

We have presented an algorithm that gives an expected 2.5-approximation guarantee for the demand matching instances where the underlying graph is bipartite and an 3-approximation expected guarantee for the demand matching instances where the underlying graph is non-bipartite. Generalizing these results to obtain a deterministic algorithm that guarantees a 2.5-approximation for bipartite and a 3-approximation for non-bipartite graphs remains open.

The work of Phillips *et al.* [PUW00] is focused on a generalization of the RESOURCE ALLOCATION PROBLEM. This is an algorithm in this area that uses LP rounding techniques. It is suggested that this work can be extended to find an approximation algorithm for demand matching which uses LP rounding; currently, no such algorithm exists. Lastly, we have demonstrated a connection between MIS and the cardinality problem and WIS and the general demand matching problem. Further, we believe that applying approximation algorithms for WIS and MIS to the cardinality problem and the demand matching problem would yield efficient approximation algorithms for these problems.

6 References

[Ber57] C. Berge. Two theorems in graph theory. In *Proceedings of the National Academy of Sciences of the U.S.A.*, 43:842– 844, 1957.

[CCHR02] G. Calinescu, A. Chakrabarti, H. J. Karloff and Y. Rabani. Improved approximation algorithms for resource allocation. In *Proceedings of the 9th International Conference on Integer Programming and Combinatorial Optimization (IPCO)*, 2337: 401–414, 2002.

[CK00] C. Chekuri and S. Khanna. A PTAS for the multiple knapsack problem. In *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 213–222, 2000.

[CMS03] C. Chekuri, M. Mydlarz and F. B. Shepherd. Multicommodity demand flow on a tree. In *Proceedings of 30th Annual International Colloquium on Automata, Languages and Pro-*

gramming (ICALP), 410–425, 2003.

[CS94] S. Cosares and I. Saniee. An Optimization Problem Related to Balancing Loads on SONET Rings. *Telecommunication Systems*, 3:165–181, 1994.

[GJ79] M. R. Garey and D. S. Johnson. Computers and intractability: a guide to the theory of NP-completeness. W. H. Freeman and Company, New York, NY, 1979.

[Hal00] M. Halldórsson. Approximation of weighted independent set and hereditary subset problems. *Journal of Graph Algorithms and Applications*, 4(1):1-16, 2000.

[Kl96] J. Kleinberg. Approximation Algorithms for Disjoint Paths Problems. PhD Thesis, Dept of EECS, MIT, 1996.

[KS02] S. G. Kolliopoukos and C. Stein. Approximation algorithms for single-source unsplittable flow. *SIAM Journal on Computing*, 31: 919–946, 2002.

[LMV00] S. Leonardi, A. Marchetti-Spaccamela and A. Vitaletti. Approximation algorithms for bandwidth and storage allocation problems under real time constraints, In *Proceedings of the 20th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, 1974:409–420, 2000.

[Lub86] M. Luby. A simple parallel algorithm for the maximal independent set problem. *SIAM Journal on Computing*, 4:1036–1053, 1986.

[MR95] R. Motwani and P. Raghavan. Randomized algorithms. Cambridge University Press, New York, NY, 1995.

[NV03] H. Q. Ngo and V. H. Vu. On multirate rearrangements Clos networks and a generatized edge colouring problem on bipartite graphs. *SIAM Journal of Computing*, 32(4):1040–1049, 2003.

[PUW00] C. A. Phillips, R. N. Uma and J. Wein. Off-line admission control for general scheduling problems. In *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA),* 879–888, 2000.

[ST93] D. Shmoys and É. Tardos. An approximation algorithm for the generalized assignment problem. *Mathematical Programming A*, 62:461–474, 1993.

[SV02] B. Shepherd and A. Vetta. The Demand Matching Problem, In *Proceedings of the 9th International Conference on Integer Programming and Combinatorial Optimization (IPCO)*, 457-476, 2002.

[SV06] B. Shepherd and A. Vetta. The Demand Matching Problem, submitted, 2006.