# FACTORS AND FLOWS

P. Hell† and D.G. Kirkpatrick‡
Technical Report 86-22
November 1986

† School of Computing Science, Simon Fraser University, Burnaby, B.C., Canada, V5A 1S6. The research of this author was supported by the Natural Sciences and Engineering Research Council grant A5075.

‡ Department of Computer Science, University of British Columbia, Vancouver, B.C., Canada, V6T 1W5. The research of this author was supported by the Natural Sciences and Engineering Research Council grant A3583.

# 1. INTRODUCTION

The *degree constrained subgraph problem* (or *$(g,f)$-factor problem*) specifies an arbitrary graph[1] $H=(V,E)$ and two functions $g$ and $f{:}\,V \rightarrow N$ and asks whether there exists a subgraph $G$ of $H$ whose degree at each vertex $v \in V$, denoted $deg_G(v)$, satisfies $g(v) \leq deg_G(v) \leq f(v)$. Such a subgraph is called a *$(g,f)$-factor* of $H$.

We consider the following generalization of the $(g,f)$-factor problem. We say that $H=(V,E,\lambda,g,f)$ is a *capacitated graph* if $(V,E)$ is a graph, $\lambda{:}E \rightarrow R^+$, and $g,f{:}\,V \rightarrow R^+$, where $g(v) \leq f(v)$, for all $v \in V$. A *packing* $\pi$ of $H$ is a function $\pi{:}E \rightarrow R$ satisfying $0 \leq \pi(e) \leq \lambda(e)$, for all $e \in E$, and $\sum_{e \ni v} \pi(e) \leq f(v)$ for all $v \in V$. Since $\sum_{e \ni v} \pi(e) \leq f(v)$ implies $\pi(uv) \leq f(v)$, for all $v \in V$, we can assume without loss of generality that $\lambda(uv) \leq \min\{f(u),f(v)\}$ for all $uv \in E$. If $\lambda(uv)=\min\{f(u),f(v)\}$ for all $uv \in E$ we say that $H$ has *unconstrained* edge capacities. Since $\sum_{e \ni v} \pi(e) \leq \sum_{e \ni v} \lambda(e)$, we can assume without loss of generality that $f(v) \leq \sum_{e \ni v} \lambda(e)$. Thus[2] $f(V) \leq 2\lambda(E)$. (For convenience we will sometimes extend the domains of $\lambda$ and $\pi$, defining $\pi(e)=\lambda(e)=0$, for $e \notin E$.) We define $deg_\pi v = \sum_{e \ni v} \pi(e)$ and $h_\pi(v) = \min\{g(v), deg_\pi v\}$. A packing $\pi$ is said to be *deficient* at vertex $v$ if $deg_\pi v < g(v)$. The *size* of the packing $\pi$, denoted $||\pi||$, is given by $||\pi|| = \sum_{v \in V} h_\pi(v)$. Note that $||\pi|| \leq g(V)$. The *degree constrained packing problem* (or *$(g,f)$-packing problem*) asks for a packing $\pi$ of $H$ that maximizes $||\pi||$. The *deficiency* of a packing $\pi$ is given by $g(V) - ||\pi||$. A packing of deficiency 0 is said to be *perfect.*

If $\lambda$, $g$, and $f$ are all integer valued, we say that $H$ is an *integer* capacitated graph. If in addition $\lambda(e)=1$ for all $e \in E$, we say that $H$ is *unit* capacitated. Similarly, if $\pi$ is integer valued (respectively, $\{0,1\}$ - valued) we say that it is an *integer* (respectively, *unit*) packing. It should be clear that the graph $H=(V,E)$ admits a $(g,f)$-factor if and only if the unit capacitated graph $H' = (V,E,\lambda,g,f)$ admits a perfect unit packing.

Amongst others, two special cases of the $(g,f)$ - packing problem, which turn out to be equivalent for integer capacitated graphs, are singled out for special attention. In the first, which we call the *$(g<f)$ - packing problem*, the degree constraints $g$ and $f$ satisfy $g(v) < f(v)$, for every $v \in V$. The second case is the *bipartite $(g,f)$ - packing problem,* that is the $(g,f)$ - packing problem restricted to bipartite graphs.

We study both the general and integer versions of the $(g,f)$-packing problem from a number of different viewpoints. First, in the spirit of Berge [6; 7, Chap. 8, Thm. 12], we present a characterization of maximum size $(g,f)$-packings in terms of the absence of certain very simple augmenting configurations. Our proof involves a new derivation of Lovasz's $(g,f)$-factor theorem using a slight extension of techniques introduced by Gallai [16].

---

[1] Unless otherwise specified, it is assumed that graphs are undirected but may have loops and multiple edges. We denote a typical edge with endpoints $v$ and $w$ as $vw$.

[2] When $X$ is a set and $f$ a function defined on $X$, we let $f(x) = \sum_{x \in X} f(x)$.

It turns out that the condition in Lovasz's $(g,f)$-factor theorem can be simplified in some cases. This was already observed by Las Vergnas [24]. We obtain such a simplification in the case

$$g(v) < f(v) \text{ or } g(v) = f(v) = 1,$$

for every $v \in V$, or the graph $H$ is bipartite. This subsumes the Las Vergnas condition

$$g(v) \leq 1,$$

for all $v \in V$, and we obtain Las Vergnas' theorem as a corollary. This also generalizes the simplification in [20], which applies when

$$g(v) < f(v),$$

for all $v \in V$, or the graph $H$ is bipartite. However, the proof given in [20] is substantially simpler.

In section 3, we set out in some detail the very close connection between the maximization problems for network flows and $(g,f)$-packings. This generalizes some of the known equivalences between network flow and bipartite matching problems. In this way the extensive literature, including characterizations and algorithmic results, on network flows can be reinterpreted in terms of certain degree constrained packings. Furthermore, this reveals some non-trivial equivalences between certain restricted $(g,f)$-packing problems.

Section 4 is concerned with the construction of an efficient algorithms for the integer $(g,f)$-packing problem. Using a variant of Gabow's algorithm [15] for finding $(g,f)$-factors with the maximum number of edges, we give an $O(\sqrt{g(V)}\lambda(E))$ upper bound for this problem.

A unifying theme that runs through this paper is our use of what we have called maximum size as our optimization criterion in the study of degree constrained packings. In section 5 we mention some other criteria. In particular, we show that the problem of finding a $(g,f)$-packing that minimizes the number of deficient vertices is, in general, NP-hard. Thus, in contrast to our earlier results, we should expect neither polynomial algorithms nor simple duality theorems for this particular optimization criterion.

A generalization of Hopcroft and Karp's bipartite matching algorithm [21] for the efficient construction of maximum size packings in integer capacitated bipartite graphs is given in the Appendix.

# 2. CHARACTERIZATIONS OF MAXIMUM (g,f)-PACKINGS

There is a rich history of characterization results for maximum $(g,f)$-packings. This includes the well known existence theorems for 1-factors (i.e. (1,1)-factors), f-factors (i.e. $(f,f)$-factors), and $(g,f)$-factors, due to Tutte [30], Tutte [31], and Lovasz [25] respectively, and the augmenting path theorems of Berge [5; 7, Chap. 8, Thm. 1] for matchings (i.e. (1,1)-packings) and $(f,f)$-packings (also called c-matchings [7]). The literature is marked by numerous simplified or alternative proofs, real and apparent generalizations, and unifications. We will restrict our discussion to only those results that bear directly on the results and techniques of this paper.

In this section, we present a characterization, in the spirit of Berge [6; 7, Chap. 8, Thm. 12], of maximum $(g,f)$-packings in terms of the absence of certain augmenting configurations. We derive new proofs of the duality results of Lovasz [25] and Las Vergnas [24].

Throughout this section $H = (V,E,\lambda,g,f)$ denotes an arbitrary capacitated graph and $\pi$ denotes some packing of $H$. An edge $e \in E$ with $\pi(e) > 0$ is said to be *used* by $\pi$. An edge $e \in E$ with $\pi(e) < \lambda(e)$ is said to be *unsaturated* by $\pi$. Note that in general an edge may be both used and unsaturated. If $X, Y \subseteq V$ we denote by $\lambda(X,Y)$ (respectively $\pi(X,Y)$) the expression $\sum_{x \in X} \sum_{y \in Y} \lambda(xy)$ (respectively $\sum_{x \in X} \sum_{y \in Y} \pi(xy)$).

We will develop our characterizations for the two cases of general and unit capacitated graphs separately. In both cases augmenting configurations are based on the familiar notion of alternating paths. An *alternating path* in $H$ with respect to $\pi$ is a sequence of not necessarily distinct vertices $v_0, v_1, \ldots, v_t \in V$ where edge $v_i v_{i+1} \in E$ is unsaturated, when $i$ is even, and used, when $i$ is odd, for $0 \le i < t$. We refer to $t$ as the *length* of the path.

## 2.1. General edge capacities

An *augmenting path* $P$ in $H$ with respect to $\pi$ has one of the following two forms:

i) $P$ is an odd length alternating path starting at a vertex $v_0$ with $deg_\pi v_0 < g(v_0)$ and ending at a vertex $v_t$ (not necessarily distinct from $v_0$) with $deg_\pi v_t < f(v_t)$; or

ii) $P$ is an even length alternating path starting at a vertex $v_0$ with $deg_\pi v_0 < g(v_0)$ and ending at a vertex $v_t$ with $deg_\pi v_t > g(v_t)$.

It should be clear that if $H$ admits an augmenting path with respect to $\pi$ then $\pi$ is not a maximum packing of $H$. ($||\pi||$ can be increased by increasing $\pi(v_i v_{i+1})$, for even $i$, and decreasing $\pi(v_i v_{i+1})$, for odd $i$, along any augmenting path. Given such a path the maximum possible such augmentation is straightforward to compute, but is not relevant for our purposes here.)

That the absence of augmenting paths is a sufficient as well as necessary condition for $\pi$ to be a maximum packing is immediate from the proof of the following theorem. If $S \subseteq V$, define $\tau(S)$ to be the set $\{v \in V - S \mid \lambda(v, V-S) < g(v)\}$.

**Theorem 2.1.** *If $\pi$ is a maximum size packing of H, then $||\pi||$ equals the minimum over all sets $S \subseteq V$, of the expression*

$$g(V - \tau(S)) + \lambda(\tau(S), V - S) + f(S)$$

**Proof.** Let $\pi$ be an arbitrary packing of $H$ and suppose that $S \subseteq V$. Let $T = \tau(S)$. Then,

$$||\pi|| = \sum_{v \in V} h_\pi(v)$$

$$\leq \sum_{v \notin T} g(v) + \sum_{v \in T} deg_\pi v$$

$$\leq g(V - T) + \sum_{v \in T} \sum_{w \in V-S} \pi(vw) + \sum_{v \in T} \sum_{w \in S} \pi(vw)$$

$$\leq g(V - T) + \lambda(T, V - S) + f(S).$$

Hence the maximum of $||\pi||$ is at most the minimum of the expression cited. To prove their equality we shall find a packing $\pi$ and a set $S$ for which

$$||\pi|| = g(V - \tau(S)) + \lambda(\tau(S), V - S) + f(S).$$

Let $\pi$ be any packing of $H$ which admits no augmenting path. We define the sets $R$ and $S$ recursively.

i) if $deg_\pi v < g(v)$ then $v \in R$;

ii) if $v \in R$, $\pi(vw) < \lambda(vw)$, and $deg_\pi w = f(w)$ then $w \in S$; and

iii) if $v \in S$, $\pi(vw) > 0$, and $deg_\pi w = g(w)$ then $w \in R$.

Since $\pi$ admits no augmenting path, it follows that $R \cap S = \phi$. In addition,

a) if $v \in S$ and $\pi(vw) > 0$, then $w \in R$

b) if $v \in R$ and $\pi(vw) < \lambda(vw)$, then $w \in S$.

Thus

$$g(V - R) + \lambda(R, V - S) + f(S)$$

$$= g(V - R) + \sum_{v \in R} \sum_{w \in V-S} \pi(vw) + \sum_{v \in R} \sum_{w \in S} \pi(vw)$$

$$= g(V - R) + \sum_{v \in R} deg_\pi v$$

$$= ||\pi||.$$

But $v \in R$ implies either $deg_{\pi} v < g(v)$ or $deg_{\pi} v = g(v)$ and there exists a $u \in S$ such $\pi(uv) > 0$. Hence, using (b) above, $\lambda(v, V-S) < g(v)$ for all $v \in R$, (i.e. $R \subseteq \tau(S)$). Furthermore, if $v \in V-(R \cup S)$ then $deg_{\pi} v \geq g(v)$ and $\pi(vw) = 0$ for all $w \in S$. Hence, $\lambda(v, V-S) \geq g(v)$ for all $v \in V-(R \cup S)$. Thus $R = \tau(S)$.

□

Note that the expression minimized in Theorem 2.1 may also be written as[3]

$$g(V) - \sum_{v \in V-S} [g(v) \dot{-} \lambda(v, V-S)] + f(S).$$

**Corollary 2.2.** *If $H$ is a capacitated graph with unconstrained edge capacities and $\pi$ is a maximum size packing of $H$, then $||\pi||$ equals the minimum over all $S \subseteq V$ of the expression $g(V-\tau(S)) + f(S)$. Furthermore, $\tau(S)$ forms an independent set in $H$.*

**Proof.** Since the edge capacities of $H$ are unconstrained, it follows from the definition of $\tau(S)$ that the elements of $\tau(S)$ are isolated vertices in $H-S$. Hence $\lambda(\tau(S), V-S) = 0$.

□

**Note.** When $H$ has unconstrained edge capacities, Corollary 2.2 implies that $H$ has a perfect packing if and only if $f(S) \geq g(\tau(S))$ for every $S \subseteq V$. This is equivalent to Theorem 2 of [8].

**Corollary 2.3.** *A packing $||\pi||$ of $H$ is maximum if and only if $H$ admits no augmenting path with respect to $\pi$.*

□

Our augmenting paths have obvious similarities with the augmenting paths that arise in the characterization of maximum flows in capacitated networks [14]. We note in Remark 3.16 that Theorem 2.1 is equivalent to the well-known max-flow min-cut theorem of Ford and Fulkerson and we suggest an easy derivation of the latter from the former.

## 2.2. Integer edge capacities

If $H = (V, E, \lambda, g, f)$ is an integer capacitated graph then we denote by $\hat{H}$ the unit capacitated graph formed from $H$ by replacing each edge $e \in E$ by $\lambda(e)$ copies of itself each with capacity 1.

**Remark 2.4.** It should be clear that the maximum integer packing of $H$ and the maximum unit packing of $\hat{H}$ have the same size. As a result characterizations and algorithms for maximum packings in unit capacitated graphs translate directly into characterizations and algorithms for maximxum packings in integer capacitated graphs.

[3] $x \dot{-} y$ denotes the expression min $\{x - y, 0\}$.

Suppose therefore that $H$ is a unit capacitated graph and that $\pi$ is a unit packing in $H$. We refer to those edges $e \in E$ for which $\pi(e) = 1$ as *dark* edges and denote them as $o = = = = = o$. Edges $e \in E$ with $\pi(e) = 0$ are referred to as *light* edges and are denoted as $o - - - - - o$. The vertex label $<g$ (respectively, $<f$, $g$, $f$, $f = g$) denotes a vertex $v$ with $deg_\pi v < g(v)$ (respectively, $<f(v)$, $=g(v)$, $=f(v)$, $=f(v) = g(v)$).

Note that we can associate with a unit packing $\pi$ the subgraph of $H$ formed by the dark edges. In this way, we can interpret results about unit $(g,f)$-packings as results about $(g,f)$-factors.

An *augmenting path* in $H$ with respect to $\pi$ is an elementary path[4] of one of the following three forms:

i) $\overset{<g}{o} - - - - [-\overset{f}{o} = = = = = \overset{g}{o}-]^{*} - - - - \overset{f}{o} = = = = = \overset{>g}{o}$,

ii) $\overset{<g}{o} - - - - [-\overset{f}{o} = = = = = \overset{g}{o}-]^{*} - - - - \overset{<f}{o}$,

where the endpoints are distinct, and

iii) $\overset{<g}{o} - - - - [-\overset{f=g}{o} = = = = = \overset{f=g}{o}-]^{*} - - - - \overset{<f-1}{o}$,

where the endpoints are identical,

where $o - - - - [-\overset{p}{o} = = = = = \overset{q}{o}-]^{*} - - - - o$ is defined recursively as $o - - - - - o$ or $o - - - - - \overset{p}{o} = = = = = \overset{q}{o} - - - - [-\overset{p}{o} = = = = = \overset{q}{o}-]^{*} - - - - o$. We note that if $H$ is bipartite or if $g(v) < f(v)$, for all $v \in V$, then augmenting paths of type iii are impossible and if $f \equiv g$ then paths of type i are impossible.

As before, it should be clear that if $H$ admits an augmenting path $P$ with respect to $\pi$ then $\pi$ is not a maximum packing of $H$. In this case an augmentation of $\pi$, increasing $||\pi||$ by at least one, is achieved by taking the symmetric difference of $\pi$ (viewed as the set of edges with $\pi(e) = 1$) and the edges of $P$. It is an immediate consequence of the proof of the following theorem that the absence of augmenting paths is also a *sufficient* condition for $\pi$ to be a maximum packing of $H$. The theorem was first proved by L. Lovasz [25] using another method. Tutte [32] has also given a different proof. As far as we know ours is the first proof of this result which uses augmenting paths explicitly. (Special cases of Lovasz' theorem have also been proved by augmenting path techniques [1],[2],[17],[18],[20],[22].)

As a preliminary to our proof of Lovasz' theorem we recall some of the basic terminology and results on augmenting paths (cf. [7],[16],[33]). Let $\pi$ be any unit packing of $H$ that admits no augmenting path. Following Berge [7] we assign colours to the vertices of $H$ in the following way:

i) if there exists an even length elementary alternating path to a vertex $w$ from a vertex $v$ with $deg_\pi v < g(v)$ then $w$ is coloured *black;*

ii) if there exists an odd length elementary alternating path to a vertex $w$ from a vertex $v$ with $deg_\pi v < g(v)$ then $w$ is coloured *white;* and

iii) if there exist both even and odd length elementary alternating paths to a vertex $w$ from a vertex $v$ with $deg_\pi v < g(v)$ then $w$ is coloured *grey.*

---

[4] A path $P$ is elementary if no edge appears more than once on $P$.

It is immediate from the definitions that,

i) if an edge $e$ joins two white vertices, then $e$ is light;
ii) if an edge $e$ joins two black vertices, then $e$ is dark;
iii) grey vertices are adjacent only to coloured vertices;
iv) if an edge $e$ joins an uncoloured vertex to a white vertex, then $e$ is light; and
v) if an edge $e$ joins an uncoloured vertex to a black vertex, then $e$ is dark.

Furthermore, by the assumption that $\pi$ admits no augmenting path, we have:

i) all black vertices $v$ have $deg_\pi v \leq g(v)$;
ii) all white vertices $v$ have $deg_\pi v = f(v)$;
iii) all grey vertices $v$ have $g(v) = f(v)$; and
iv) all uncoloured vertices $v$ have $g(v) \leq deg_\pi v \leq f(v)$.

Let Black$_\pi$ (respectively, White$_\pi$, Grey$_\pi$) denote the set of black (respectively, white, grey) vertices of $H$. Let $C$ be any ***grey component*** - i.e. a connected component of $H|\text{Grey}_\pi$. It is a direct consequence of [7,16] that $C$ is of one of the following three types:

*type A* - $C$ has a unique vertex $v$ with $deg_\pi v = g(v) - 1$. All other vertices $w \in C$ have $deg_\pi w = g(w)$. All edges joining $C$ to white vertices are light and all edges joining $C$ to black vertices are dark.

*type B* - All vertices $w \in C$ have $deg_\pi w = g(w)$. All but exactly one edge joining $C$ to white vertices are light and all edges joining $C$ to black vertices are dark.

*type C* - All vertices $w \in C$ have $deg_\pi w = g(w)$. All edges joining $C$ to white vertices are light and all but exactly one edge joining $C$ to black vertices are dark.

Note that every grey component has size greater than one and if $v \in \text{Grey}_\pi$ then

$$\lambda(v, V\text{--White}_\pi) \geq \begin{cases} 1+deg_\pi v & v \text{ is type A} \\ deg_\pi & v \text{ is type B} \\ 1+deg_\pi v & v \text{ is type C} \end{cases}$$

$$\geq g(v).$$

Furthermore if $C$ is any grey component, then

$$f(C)+\lambda(C,\text{Black}_\pi) = 1+\sum_{v\in C}\sum_{w\in C}\pi(vw)+2\sum_{v\in C}\sum_{w\in\text{Black}_\pi}\pi(vw).$$

$$\equiv 2(mod\ 2)$$

Let $S$ and $T$ be disjoint subsets of $V$. Let $\Xi(H,S,T)$ denote the set of connected components $C$ of $H-(S\cup T)$ with the property that $f(x) = g(x)$ for all $x \in C$ and $f(C) + \lambda(C,T) \equiv 1(mod\ 2)$.

**Theorem 2.5**. *If $H$ is integer capacitated and $\pi$ is a maximum size integer packing of $H$, then $||\pi||$ equals the minimum, over all pairs of disjoint sets $S,T$ of the expression*

$$g(V-T) + f(S) + \lambda(V-S,T) - |\Xi(H,S,T)|.$$

**Proof**. By Remark 2.4 it suffices to prove the result for unit capacitated graphs. Let $\pi$ be an arbitrary unit packing of $H$. Suppose $S,T \subseteq V$ are disjoint and $\Xi = \Xi(H,S,T)$. Then,

$$||\pi|| = \sum_{v\in V} h_\pi(v)$$

$$\leq \sum_{v\notin T} h_\pi(v) + \sum_{v\in T} deg_\pi v$$

$$\leq g(V-T) - \sum_{C\in\Xi}\sum_{v\in C}(f(v) - h_\pi(v)) + \sum_{v\in T} deg_\pi v$$

$$= g(V-T) - \sum_{C\in\Xi}\sum_{v\in C}(f(v) - h_\pi(v)) + \sum_{v\in T}\sum_{w\in S}\pi(vw) + \sum_{v\in T}\sum_{w\in V-S}\pi(vw)$$

$$\leq g(V-T) - \sum_{C\in\Xi}\sum_{v\in C}(f(v) - h_\pi(v)) + f(S) - \sum_{v\in V-T}\sum_{w\in S}\pi(vw) + \sum_{v\in T}\sum_{w\in V-S}\pi(vw)$$

$$= g(V-T) + f(S) + \lambda(V-S,T)$$

$$- \sum_{C\in\Xi}\sum_{v\in C}(f(v) - h_\pi(v)) - \sum_{v\in V-T}\sum_{w\in S}\pi(vw) - \sum_{v\in T}\sum_{w\in V-S}(\lambda(vw) - \pi(vw))$$

$$\leq g(V-T) + f(S) + \lambda(V-S,T)$$

$$- \sum_{C\in\Xi}\left[\sum_{v\in C}(f(v) - h_\pi(v)) + \sum_{v\in T}\sum_{w\in C}(\lambda(vw) - \pi(vw)) + \sum_{v\in C}\sum_{w\in S}\pi(vw)\right]$$

$$\leq g(V-T) + f(S) + \lambda(V-S,T) - \sum_{C\in\Xi} 1,$$

since at least one of the sums is non-zero,

$$= g(V-T) + f(S) + \lambda(V-S,T) - |\Xi(H,S,T)|.$$

Hence the maximum of $||\pi||$ is at most the minimum of the expression cited. To prove equality, let $\pi$ be any unit packing of $H$ that admits no augmenting path and let $S$=White$_\pi$ and $T$=Black$_\pi$. It follows from the definitions that every component of

$H|\text{Grey}_\pi$ belongs to $\Xi(H,S,T)$. Hence,

$$|\Xi(H,S,T)| \geq |\Xi_A(H,S,T)| + |\Xi_B(H,S,T)| + |\Xi_C(H,S,T)|\ ,$$

where $\Xi_A(H,S,T)$ (respectively, $\Xi_B(H,S,T)$, $\Xi_C(H,S,T)$) denotes the set of components of $H|\text{Grey}_\pi$ of type A (respectively, type B, type C). Thus,

$$\begin{aligned}
||\pi|| &= h_\pi(V-T) + \sum_{v\in T} deg_\pi v \\
&= h_\pi(V-T) + \sum_{v\in T}\sum_{w\in S}\pi(vw) + \sum_{v\in T}\sum_{w\in V-S}\pi(vw) \\
&= g(V-T) - |\Xi_A(H,S,T)| + f(S) - |\Xi_B(H,S,T)| + \lambda(V-S,T) - |\Xi_C(H,S,T)| \\
&\geq g(V-T) + f(S) + \lambda(V-S,T) - |\Xi(H,S,T)|.
\end{aligned}$$

□

**Corollary 2.6.** *An integer packing $\pi$ of $H$ has maximum size if and only if $\hat{H}$ admits no augmenting path with respect to $\pi$.*

□

This statement is further quantified in lemma 4.1 below.

Lovasz's condition in Theorem 2.5 is quantified over two disjoint subsets $S$ and $T$ of the vertices of $H$. In some cases the condition may be simplified, and stated in terms of only *one* subset $S$. A well known example of this kind of simplification occurs in Tutte's theorem for (1,1)-factors [30]; this was generalized by Las Vergnas to $(g,f)$-factors with $g(v) \leq 1$, for all $v \in V$. We now give such a simplification for the case when $g(v) < f(v)$ or $g(v) = f(v) = 1$, for all $v \in V$. Let $H$ be any integer capacitated graph and let $S \subseteq V$. We denote by $\Xi'(H,S)$ the set of connected components $C$ of $H - S$ with $|C| \geq 3$ and odd and $g(v) = f(v) = 1$, for all $v \in C$. Recall that $\tau(S) = \{v \in V{-}S \mid \lambda(v,V{-}S) < g(v)\}$.

**Theorem 2.7.** *If $\pi$ is a maximum size integer packing of the integer capacitated graph $H$ with $g(v) < f(v)$ or $g(v) = f(v) = 1$, for all $v \in V$, then $||\pi||$ equals the minimum, over all $S \subseteq V$, of the expression*

$$g(V{-}\tau(S)) + f(S) + \lambda(V{-}S,\tau(S)) - |\Xi'(H,S)|.$$

**Proof.** As before it suffices to prove the result for unit capacitated graphs.

Let $\pi$ be an arbitrary unit packing of $H$ and suppose $S \subseteq V$. If $C \in \Xi'(H,S)$ and $v \in C$ then $g(v) = f(v) = 1$ and $\lambda(v,V{-}S) > 0$, and hence $v \notin \tau(S)$. It follows that $\lambda(C,\tau(S)) = 0$ and $f(C) + \lambda(C,\tau(S)) \equiv 1 (mod\ 2)$, that is $C \in \Xi(H,S,\tau(S))$. Thus

$$||\pi|| \leq g(V-\tau(S)) + f(S) + \lambda(V-S,\tau(S)) - |\Xi'(H,S)|$$

is an immediate consequence of Theorem 2.5.

To establish equality, let $\pi$ be any unit packing of $H$ that admits no augmenting path. Let $S=\text{White}_\pi$, $T=\text{Black}_\pi$ and $R=\text{Grey}_\pi$. First note that, with the additional assumptions on $g$ and $f$, it follows that no edge of $H$ joins any vertex in $T$ to any vertex $v \in R$. (Such a vertex would have $deg_\pi v > 1$). Hence, for each $C \in \Xi(H,S,T)$, $\lambda(C,T) = 0$ and so $f(C) + \lambda(C,T) = |C|$. Thus $\Xi(H,S,T) \subseteq \Xi'(H,S)$. Furthermore, for each $v \in T$, $\lambda(v,V-S) = deg_\pi v - \sum_{w\in S} \lambda(vw) < g(v)$. That is, $T \subseteq \tau(S)$. But, if $v \in \tau(S)$, then either $v$ is isolated in $H - S$ or $g(v) > 1$, that is $v \notin R$. It follows that $\tau(S) \subseteq T$, and hence $T = \tau(S)$. So, by the proof of Theorem 2.5,

$$||\pi|| = g(V-\tau(S)) + f(S) + \lambda(V-S,\tau(S)) - |\Xi'(H,S)|.$$

□

Note that the expression minimized in Theorem 2.7 can also be written as

$$g(V) - \sum_{v\in V-S} [g(v) \dot{-} deg_{H-S}(v)] + f(S) - |\Xi'(H,S)|.$$

We now derive the simplification due to Las Vergnas [24]:

**Corollary 2.8.** *If $\pi$ is a maximum size integer packing of the integer capacitated graph $H$ with $g(v) \in \{0,1\}$, for all $v \in V$, then $||\pi||$ equals the minimum, over all $S \subseteq V$, of the expression*

$$g(V-I_S) + f(S) - |\Xi'(H,S)|,$$

*where $I_S$ denotes the set of isolated vertices in $H - S$.*

**Proof.** In light of Theorem 2.7, it suffices to note that if $g(v) = 1$ for all $v \in V$, then for any set $S \subseteq V$, $\tau(S) \subseteq I_S$, and $g(I_S-\tau(S)) = 0$.

□

In [20] we give a simple proof (together with applications) of the following:

**Corollary 2.9.** *If $\pi$ is a maximum size integer packing of the integer capacitated graph $H$, where $H$ is bipartite or has $g(v) < f(v)$ for all $v \in V$, then $||\pi||$ equals the minimum, over all sets $S \subseteq V$, of the expression*

$$g(V) - \sum_{v\in V-S} [g(v) \dot{-} deg_{H-S}(v)] + f(S).$$

**Proof.** In the case that $g(v) < f(v)$, for all $v \in V$, the result follows directly from Theorem 2.7. When $H$ is bipartite the proof is similar to that of Theorem 2.1 and will not be repeated here.

□

**Note.** Comparing Theorem 2.7 and its corollaries with Theorem 2.5 (Lovasz's theorem), we note that the simplification in these special cases is due to the fact that the set $T$ can be fixed as $\tau(S)$.

It follows from Theorem 2.1 and Corollary 2.9 that if the integer capacitated graph $H$ is bipartite or if $g(v) < f(v)$ for all $v \in V$, then there exists a maximum size packing that is an integer packing.

# 3. (g,f)-PACKING AND NETWORK FLOW PROBLEMS.

In this section we explore the close connection between the maximization problems for network flows and $(g,f)$-packings. Specifically, we show that a number of variants of the $(g,f)$-packing problem are equivalent to the well-studied maximum flow problem. As a byproduct, we describe upper bounds for variants of $(g,f)$-packing based on efficient network flow algorithms. In section 4, we consider other algorithms for integer $(g,f)$-packings.

## 3.1. Definitions and notation

As in the preceding section, $H = (V,E,\lambda,g,f)$ denotes a capacitated graph and $\pi$ denotes an arbitrary packing of $H$. We say that $N = (X,A,c)$ is a ***network*** if $(X,A)$ is a directed graph with two distinguished vertices, a ***source*** $s$ and a ***sink***, and $c{:}A \rightarrow R^+$. We call $c(a)$ the ***capacity*** of arc $a$. (For convenience we define $c(a) = 0$ for $a \notin A$ and, as usual, we abbreviate $c((x,y))$ by $c(x,y)$.) Vertices in $X_I = X - \{s,t\}$ are said to be ***internal*** vertices of $N$.

A ***flow*** in the network $N$ is a function $\xi{:}A \rightarrow R$ satisfying $0 \leq \xi(a) \leq c(a)$, for all $a \in A$, and $\xi(X,x) = \xi(x,X)$ for all $x \in X_1$. The ***value*** of the flow $\xi$, denoted $|\xi|$, is defined to be the flow out of $s$, namely $\xi(s,X)$. The ***maximum flow problem*** asks for a flow $\xi$ in $N$ that maximizes $|\xi|$.

The arc capacities constrain from above the flow that can pass through an arbitrary vertex $x$. We define the ***vertex capacity*** of $x$, denoted $C(x)$, to be

$$C(x) = \begin{cases} \min\{c(X,x),c(x,X)\} & x \in X_I \\ \min\{c(X,t),c(s,X)\} & x \in \{s,t\}. \end{cases}$$

Note that $C(X) \leq 2c(A)$. It should also be clear that $\xi(x,X) \leq C(x)$, for all $x \in X_I$. A network is said to be of ***type 2*** (cf. [13]) if $c(a) = 1$ for all $a \in A$ and $C(x) = 1$ for all $x \in X_I$.

We will also be interested in the case where networks have non-zero lower bounds $b(a)$ on the permissible flow through edges $a \in A$. In this case a ***feasible*** flow $\xi$ is defined to be a flow which satisfies $\xi(a) \geq b(a)$, for all $a \in A$. These lower bounds constrain from below the permissible flow through vertices in the obvious way. We define the ***vertex demand*** of $x$, denoted $B(x)$ as

$$B(x) = \begin{cases} \max\{b(X,x),b(x,X)\} & x \in X_I \\ \max\{b(X,t),b(s,X)\} & x \in \{s,t\} \end{cases}$$

In our discussions of the relative and absolute complexity of packing and network flow problems we adopt the (familiar) assumption of a unit-charged random access machine as an underlying model of computation. On this model it is natural to define $|E|$ as the ***size*** of a capacitated graph $H = (V,E,\lambda,g,f)$ and define $|A|$ as the ***size*** of the network $N = (X,A)$. It should be noted that these definitions are based on the (possibly unrealistic) assumption that edge capacities can each be represented in one memory location. On a logarithmically charged model, we might redefine size as $\sum_{e \in E} (1+log\lambda(e))$ and $\sum_{a \in A} (1+logc(a))$ respectively. The results of this section are independent of these considerations. Our reductions are all *linear* in that that take instances of one problem A of size $n$ into instances of another problem B of size $O(n)$ and transform a solution to problem B back to a solution to problem A in $O(n)$ time. Here size can be defined in either of the ways above (or in any reasonably variant of these). We will say that two problems are ***linearly equivalent*** if there exist linear reductions taking each to the other.

## 3.2. Bipartite (f,f)-packing and network flow

In this section, we consider the very close connection between the maximum flow problem and the *bipartite $(f,f)$-packing problem,* that is the special case of bipartite packing in which $g(v) = f(v)$, for all $v \in V$. The results here are all in the spirit of the folklore on this topic. We review them here simply to set the stage for the more general reductions that follow.

Let $H = (V,E,\lambda,f,f)$ be a capacitated bipartite graph with bipartition $V = U \cup W$. Consider the network $N = (X,A,c)$ constructed from $H$ as follows:

$$X = \{v' | v \in V\} \cup \{s,t\}$$

$$A = \{(s,u') | u \in U\} \cup \{(w',t) | w \in W\} \cup \{(u',w') | uw \in E\}$$

and $c(s,u') = f(u)$, $c(w',t) = f(w)$, and $c(u',w') = \lambda(uw)$.

Note that $|X| = |V| + 2$, $|A| = |V| + |E|$, $c(A) = f(V) + \lambda(E) \leq 3\lambda(E)$ and $C(X) \leq 2f(V)$. Furthermore, if $H$ is integer capacitated then so is $N$.

**Lemma 3.1.** *$H$ admits a packing $\pi$ of size $z$ if and only if $N$ admits a flow $\xi$ of value $z/2$.*

**Proof.** Let $\pi$ be a packing of $H$. Then $N$ admits a flow $\xi$ of size $||\pi||/2$ by assigning: $\xi(u',w') = \pi(uw)$, $\xi(s,u') = \sum_{w \in W} \pi(uw)$, and $\xi(w,t) = \sum_{u \in U} \pi(uw)$. The inverse of the same transformation translates a flow $\xi$ in $N$ to a packing $\pi$ of $H$ of size exactly $2|\xi|$.

$\square$

Let $N = (X,A,c)$ be any network. Consider the capacitated bipartite graph $H = (V,E,\lambda,f,f)$ constructed from $N$ as follows:

$$V = \{x_{in}, x_{out} | x \in X_I\} \cup \{s_{out}, t_{in}\},$$

$$E = \{x_{in}x_{out} | x \in X_I\} \cup \{x_{out}y_{in} | (x,y) \in A\},$$

$$f(s_{out}) = f(t_{in}) = C(s),$$

$$f(x_{in}) = f(x_{out}) = \lambda(x_{in}x_{out}) = C(x), \quad \text{for all } x \in X_I, \text{ and}$$

$$\lambda(x_{out}y_{in}) = c(x,y), \text{ for all } (x,y) \in A.$$

Note that $|V| = 2|X| - 2$, $|E| = |A| + |X| - 2$, $f(V) \leq 2C(X)$, and $\lambda(E) \leq c(A) + C(X) \leq 3c(A)$. Furthermore, if $N$ is integer capacitated then so is $H$.

**Lemma 3.2.** *$N$ admits a flow $\xi$ of value $z$ if and only if $H$ admits a packing $\pi$ of size $f(V) + 2(z - C(s))$.*

**Proof.** Let $\xi$ be a flow in $N$ and let $\pi$ be the packing of $H$ given by

$$\pi(x_{out}y_i) = \xi(x,y), \text{ for } (x,y) \in A, \text{ and}$$

$$\pi(x_{in}x_{out}) = \lambda(x_{in},x_{out}) - \xi(X,x), \text{ for } x \in X_I.$$

Then $||\pi|| = 2|\xi| + 2C(X_I) = f(V) + 2(|\xi| - C(s))$. Note that the packing $\pi$ saturates all of the vertices of $H$ except possibly $s_{out}$ and $t_{in}$. Any such packing is said to be a *normal* packing of $H$. To prove the converse we observe that if $\pi$ is any normal packing of $H$ then the inverse of the transformation above yields a flow $\xi$ satisfying $||\pi|| = f(V) + 2(|\xi| - C(s))$. Thus it suffices to show that for any packing $\pi$ of $H$ there is a normal packing $\pi'$ with $||\pi'|| \geq ||\pi||$.

Let $\pi'$ be a packing, among packings of size at least $||\pi||$, that maximizes the total assignment to edges of the form $x_{in}x_{out}$. Suppose that some vertex $x_{in}$ is not saturated; the case when a vertex $x_{out}$ is not saturated is handled by a symmetric argument. By the maximality of $\pi'$ we know that $x_{out}$ must be saturated. Since $f(x_{in}) = f(x_{out})$ it follows that $\pi(x_{out}y_{in}) > 0$ for some $x_{out}y_{in} \in E$ with $x \neq y$. Let $\Delta = \min\{f(x_{in}) - deg_{\pi}x_{in}, \pi(x_{out}y_{in})\}$. Then the packing $\pi'$ defined by

$$\pi'(e) = \begin{cases} \pi(e) + \Delta & e = x_{in}x_{out} \\ \pi(e) - \Delta & e = x_{out}y_{in} \\ \pi(e) & otherwise \end{cases}$$

has $||\pi'|| = ||\pi||$ and a greater total assignment to edges of the form $x_{in}x_{out}$, contradicting the maximality of $\pi$.

$\square$

The reductions of Lemmas 3.1 and 3.2 combine to demonstrate the following.

**Theorem 3.3.** *The maximum flow problem on capacitated (respectively, integer capacitated) networks is linearly equivalent to the bipartite (f,f)-packing problem on capacitated (respectively, integer capacitated) graphs.*

□

Motivated by Theorem 3.3, we observe that the upper bound of Even and Tarjan [13] on the complexity of finding maximum flows in type 2 networks is easily generalized to the following.

**Theorem 3.4.** *For any integer network $N = (X,A)$, Dinic's algorithm finds a maximum flow in at most $O(\sqrt{C(X)}\,c(A))$ steps.*

**Proof.** Each edge $a \in A$ can be replaced by $c(a)$ edges each of capacity 1. The resulting unit capacitated network has a total of $c(A)$ edges. A maximum flow of value $M$ in this network can be decomposed into $M$ edge disjoint directed paths from $s$ to $t$ (plus some directed cycles). A given vertex $x$ sits on at most $C(x)$ of these paths. If the shortest of these paths has length $l$ then $M(l+1) \leq C(X)$ or $l < C(X)/M$.

The remainder of the proof is completely analogous to Theorem 3 of [13].

□

**Corollary 3.5.** *There exists an $O(\sqrt{f(V)}\lambda(E))$ algorithm for finding a maximum integer packing of the capacitated bipartite graph $H = (V,E,\lambda,f,f)$.*

□

**Remark 3.6.** This is the same bound achieved by Gabow [15] for finding a $(0,f)$-factor with the maximum number of edges (equivalently, an $(f,f)$-packing of maximum size), even when the graph is not restricted to being bipartite. In this generality, however, Gabow's algorithm depends on a non-trivial reduction to general matching and the (inherent) complications of blossoms.

The transformations above are easily modified to accommodate lower as well as upper bounds on the edge capacities of a network. In this more general setting we must be concerned with the question of feasibility as well as optimality of flows. Let $N = (X,A,c,b)$ be any network with both lower and upper bounds on the capacity of its edges. We will assume, with no loss of generality, that $(s,t) \notin A$.

Consider the capacitated bipartite graph $H_F = (V,E,\lambda,f,f)$ constructed from $N$ as follows:

$$V = \{x_{in}, x_{out} | x \in X_I\} \cup \{s_{out}, t_{in}\},$$

$$E = \{x_{out}x_{in} | x \in X_I\} \cup \{x_{out}y_{in} | (x,y) \in A\} \cup \{s_{out}t_{in}\},$$

$$f(x_{out}) = C(x) - b(x,X), \quad \text{for } x \in X - \{t\},$$

$$f(x_{in}) = C(x) - (X,x), \quad \text{for } x \in X - \{s\},$$

$$\lambda(x_{out}x_{in}) = C(x) - B(x),\ \lambda(x_{out}y_{in}) = c(x,y) - b(x,y), \text{ and } \lambda(s_{out}t_{in}) = C(s) - B(s).$$

**Lemma 3.7.** *$N$ admits a feasible flow if and only if $H_F$ admits a perfect packing.*

**Proof.** Suppose that $\xi$ is a feasible flow in $N$. In particular, $b(x,y) \leq \xi(x,y) \leq c(x,y)$ for all $(x,y) \in A$. Let $\pi$ be the packing of $H_F$ given by

$$\pi(x_{out}y_{in}) = \xi(x,y) - b(x,y), \text{ for } (x,y) \in A,$$

$$\pi(x_{out}x_{in}) = C(x) - \xi(x,X), \text{ for } x \in X_I, \text{ and}$$

$$\pi(s_{out}t_{in}) = C(s) - \xi(s,X) = C(t) - \xi(X,t).$$

Then

$$deg_\pi x_{out} = \pi(x_{out}x_{in}) + \sum_{y \in X} \pi(x_{out}y_{in})$$

$$= (C(x) - \xi(x,X)) + (\xi(x,X) - b(x,X))$$

$$= f(x_{out})$$

$$deg_\pi x_{in} = \pi(x_{out}x_{in}) + \sum_{y \in X} \pi(y_{out}x_{in})$$

$$= (C(x) - \xi(X,x)) + (\xi(X,x) - b(X,x))$$

$$= f(x_{in})$$

for $x \in X_I$, and by exactly the same arguments (reading $t_{in}$ for $s_{in}$ and $s_{out}$ for $t_{out}$) $deg_\pi s_{out} = f(s_{out})$ and $deg_\pi t_{in} = f(t_{in})$. Hence $||\pi|| = f(V)$, that is $\pi$ is a perfect packing of $H_F$.

Conversely, if $H_F$ admits a perfect packing $\pi$ then consider the function $\xi$ given by $\xi(x,y) = \pi(x_{out}y_{in}) + b(x,y)$ for all $(x,y) \in A$. Clearly $b(x,y) \leq \xi(x,y) \leq c(x,y)$. Furthermore, for every $x \in X_I$

$$\xi(X,x) = \sum_{y \in X} \pi(y_{out}x_{in}) + b(X,x)$$

$$= f(x_{in}) - \pi(x_{out}x_{in}) + b(X,x)$$

$$= C(x) - \pi(x_{out}x_{in})$$

$$= f(x_{out}) - \pi(x_{out}x_{in}) + b(x,X)$$

$$= \sum_{y \in X} \pi(x_{out}y_{in}) + b(x,X)$$

$$= \xi(x,X).$$

That is $\xi$ is a feasible flow in $N$.

□

Suppose that we modify the construction of $H_F$ by making $\lambda(s_{out}t_{in}) = 0$. Call the resulting capacitated bipartite graph $H_{\max}$.

**Lemma 3.8.** *$N$ admits a maximum flow $\xi$ of value $z$ if and only if $H_{\max}$ admits a maximum packing $\pi$ of size $f(V) - 2(C(s) - z)$.*

**Proof.** Let $\xi$ be a maximum feasible flow in $N$ and define $\pi$ to be the packing of $H_{\max}$ given by

$$\pi(x_{out}y_{in}) = \xi(x,y) - b(x,y), \text{ for } (x,y) \in A,$$

$$\pi(x_{out}x_{in}) = C(x) - \xi(x,X), \text{ for } x \in X_I, \text{ and}$$

$$\pi(s_{out}t_{in}) = 0.$$

Then $deg_\pi x_{out} = f(x_{out})$ and $deg_\pi x_{in} = f(x_{in})$ for every $x \in X_I$ as in Lemma 3.7. Furthermore,

$$deg_\pi t_{in} = \sum_x \pi(x_{out}t_{in})$$

$$= \xi(X,t) - b(X,t)$$

$$= f(t_{in}) - (C(s) - |\xi|)$$

Similarly, $deg_\pi s_{out} = f(s_{out}) - C(s) - |\xi|)$. Hence, $||\pi|| = f(V) - 2C(s) - |\xi|)$.

Conversely, we know that there exists a maximum size packing $\pi$ of $H_{\max}$ in which all vertices except possibly $s_{out}$ and $t_{in}$ are saturated and $deg_\pi t_{in} - deg_\pi s_{out} = b(s,X) - b(X,t)$. Such a packing is constructed by starting with any perfect packing of $H_F$ (cf. Lemma 3.7) in which $\pi(s_{out}t_{in})$ has been set to 0 and improving to optimality by a sequence of augmentations (all of which start at $s_{out}$ and end at $t_{in}$; cf. Section 2.1). This packing suggests the function $\xi$ where $\xi(x,y) = \pi(x_{out}y_{in}) + b(x,y)$ for all $(x,y) \in A$. Clearly, $b(x,y) \leq \xi(x,y) \leq c(x,y)$ for all $(x,y) \in A$ and, as in Lemma 3.7, $\xi(X,x) = \xi(x,X)$, for all $x \in X_I$. That is, $\xi$ is a feasible flow in $N$. Furthermore,

$$2|\xi| = \xi(s,X) + \xi(X,t)$$

$$= (deg_\pi s_{out} + b(s,X)) + (deg_\pi t_{in} + b(X,t))$$

$$= (deg_\pi s_{out} + C(s) - f(s_{out})) + (deg_\pi t_{in} + C(t) - f(t_{in}))$$

$$= ||\pi|| - f(V) + 2C(s).$$

$\square$

Suppose that we modify the construction of $H_F$ by adding two new vertices $s_{in}$ and $t_{out}$ with $f(s_{in}) = f(t_{out}) = C(s) - B(s)$ and setting $\lambda(s_{out}s_{in}) = \lambda(t_{out}t_{in}) = C(s) - B(s)$ and $\lambda(s_{out}t_{in}) = 0$. Call the resulting capacitated bipartite graph $H_{\min}$.

**Lemma 3.9.** *$N$ admits a minimum flow $\xi$ of value $z$ if and only if $H_{\min}$ admits a maximum packing $\pi$ of size $f(V) + 2(B(s) - z)$.*

**Proof.** Let $\xi$ be a maximum feasible flow in $N$ and define $\pi$ to be the packing of $H_{\min}$ given by

$$\pi(x_{out}y_{in}) = \xi(x,y) - b(x,y), \text{ for } (x,y) \in A,$$

$$\pi(x_{out}x_{in}) = C(x) - \xi(x,X), \text{ for } x \in X_I,$$

$$\pi(s_{out}s_{in}) = \pi(t_{out}t_{in}) = C(s) - |\xi|, \text{ and}$$

$$\pi(s_{out}t_{in}) = 0.$$

Then $deg_\pi v = f(v)$ as before, for every $v \in V - \{s_{in}, t_{out}\}$. Furthermore,

$$deg_\pi s_{in} = \pi(s_{out}s_{in})$$

$$= C(s) - |\xi|$$

$$= f(s_{in}) + B(s) - |\xi|.$$

Similarly, $deg_\pi t_{out} = f(t_{out}) + B(t) - |\xi|$. Hence, $||\pi|| = f(V) + 2B(s) - |\xi|)$.

Conversely, we know that there exists a maximum size packing $\pi$ of $H_{\min}$ in which $deg_\pi v = f(v)$ for all $v \in V - \{s_{in}, t_{out}\}$. Such a packing is constructed by starting with any perfect packing of $H_F$ in which $\pi(s_{out}s_{in})$ and $\pi(t_{out}t_{in})$ have been set to $\pi(s_{out}t_{in})$ and $\pi(s_{out}t_{in})$ has been set to 0. This packing can be improved to optimality by a sequence of augmentations all of which start at $s_{in}$ and terminate at $t_{out}$ (cf. Section 2.1). The resulting packing suggests the function $\xi$ where $\xi(x,y) = \pi(x_{out}y_{in}) + b(x,y)$ for all $(x,y) \in A$. Clearly, $b(x,y) \leq \xi(x,y) \leq c(x,y)$ for all $(x,y) \in A$ and, as before, $\xi(X,x) = \xi(x,X)$, for all $x \in X_I$. That is, $\xi$ is a feasible flow in $N$. Furthermore,

$$2|\xi| = \xi(s,X) + \xi(X,t)$$

$$= (deg_\pi s_{out} - deg_\pi s_{in} + b(s,X)) + (deg_\pi t_{in} - deg_\pi t_{out} + b(X,t))$$

$$= C(s) - deg_\pi s_{in} + C(t) - deg_\pi t_{out}$$

$$= f(V) - ||\pi|| + 2B(s)$$

$\square$

Lemmas 3.7 through 3.9 combine to prove the following.

**Theorem 3.10.** *The feasibility problem for flows in capacitated (respectively, integer capacitated) networks with edge demands and capacities is linearly reducible to the perfect bipartite $(f,f)$-packing problem on capacitated (respectively, integer capacitated) graphs. Furthermore, the problem of maximizing or minimizing the feasible flow in capacitated (respectively, integer capacitated) networks with edge demands and capacities is linearly reducible to the bipartite $(f,f)$-packing problem on capacitated (respectively, integer capacitated) graphs.*

## 3.3. Bipartite (g,f)-packing and network flow

Let $H = (V,E,\lambda,g,f)$ be a capacitated bipartite graph with bipartition $V = U \cup W$. Consider the network $N = (X,A,c)$ constructed from $H$ as follows:

$$X = \{\dot{u},\ddot{u} | u \in U\} \cup \{\dot{w},\ddot{w} | w \in W\} \cup \{s,t\},$$

$$A = \{(s,\dot{u}),(\ddot{u},\dot{u}),(\ddot{u},t) | u \in U\} \cup \{(s,\ddot{w}),(\dot{w},\ddot{w}),(\dot{w},t) | w \in W\}$$

$$\cup \{(\dot{u},\dot{w}),(\ddot{w},\ddot{u}) | uw \in E\}$$

and

$$c(s,\dot{u}) = c(\ddot{u},t) = g(u),\ c(s,\ddot{w}) = c(\dot{w},t) = g(w),$$

$$c(\ddot{u},\dot{u}) = f(u) - g(u),\ c(\dot{w},\ddot{w}) = f(w) - g(w), \text{ and}$$

$$c(\dot{u},\dot{w}) = c(\ddot{w},\ddot{u}) = \lambda(uw).$$

Note that $|X| = 2|V| + 2$, $|A| = 3|V| + 2|E|$, $c(A) = 2f(V) + 2\lambda(E) \leq 6\lambda(E)$, and $C(X) \leq 2f(V) + 2g(V) \leq 4f(V)$. Furthermore, if $H$ is integer capacitated then so is $N$.

**Lemma 3.11.** *$H$ admits a packing $\pi$ of size $z$ if and only if $N$ admits a flow $\xi$ of value $z$.*

**Proof.** Let $\pi$ be a packing of $H$. Then $N$ admits a flow $\xi$ of value $||\pi||$ by assigning $\xi(\dot{u},\dot{w}) = \xi(\ddot{w},\ddot{u}) = \pi(uw)$, $\xi(s,\dot{u}) = \xi(\ddot{u},t) = h_\pi(u)$, $\xi(s,\ddot{w}) = \xi(\dot{w},t) = h_\pi(w)$, $\xi(\ddot{u},\dot{u}) = deg_\pi - h_\pi(u)$, and $\xi(\dot{w},\ddot{w}) = deg_\pi w - h_\pi(w)$.

Conversely, let $\xi$ be any flow in $N$. We know that $|\xi| = \xi(s,\dot{u}) + \xi(s,\ddot{w}) = \xi(\ddot{u},t) + \xi(\dot{w},t)$. Assume, without loss of generality, that $\xi(s,\dot{u}) + \xi(\dot{w},t) \geq |\xi|$. Then, the packing $\pi$ defined by $\pi(uw) = \xi(\dot{u},\dot{w})$ has the property that $h_\pi(u) \geq \xi(s,\dot{u})$ and $h_\pi(w) \geq \xi(\dot{w},t)$ and hence $||\pi|| \geq |\xi|$.

$\square$

**Remark 3.12.** The transformation of Lemma 3.11 provides a linear reduction from the (integer) bipartite $(g,f)$-packing problem to the (integer) network flow problem. It follows, from Theorem 3.4, that there exists an $O(\sqrt{f(V)}\lambda(E))$ algorithm for the integer bipartite $(g,f)$-packing problem. In light of Theorem 3.3, it also follows that the integer bipartite $(g,f)$-packing problem is linearly reducible to the integer bipartite $(f,f)$-packing problem. Gabow [15] provides another such transformation that holds even for the non-bipartite varients of these problems. Using this he is able to derive a $O(\sqrt{f(V)}\lambda(E))$ upper bound for the integer $(g,f)$-packing problem (cf. Remark 3.6). We look more carefully at this problem in the section 4 and observe that this upper bound can be tightened when $g(V)$ is $o(f(V))$.

## 3.3.2. Network flow to bipartite (g<f)-packing

Let $N = (X,A,c)$ be any capacitated network. Consider the capacitated bipartite graph $H = (V,E,\lambda,g,f)$ constructed from $N$ as follows:

$$V = \{x_{in},x_{out} | x \in X_I\} \cup \{s_{out},t_{in}\},$$

$$E = \{x_{in}x_{out} | x \in X_I\} \cup \{x_{out}y_{in} | (x,y) \in A\},$$

$$g(x_{in}) = f(x_{in}) - 1 = g(x_{out}) + 1 = f(x_{out}) = \lambda(x_{in}x_{out}) = C(x), \text{ for all } x \in X_I,$$

$$g(s_{out}) = 0 \text{ and } f(s_{out}) = g(t_{in}) = f(t_{in}) - 1 = C(s), \text{ and}$$

$$\lambda(x_{out}y_{in}) = c(x,y), \text{ for all } (x,y) \in A.$$

Note that $|V| = 2|X| - 2$, $|E| = |A| + |X| - 2$, $g(v) < f(v)$, for all $v \in V$, $f(V) \leq 2C(X)$, and $\lambda(E) \leq c(A) + C(X) \leq 3c(A)$. Furthermore, if $N$ is integer capacitated then so is $H$.

**Lemma 3.13.** *$N$ admits a flow $\xi$ of value $z$ if and only if $B$ admits a packing $\pi$ of size $g(V) + z - C(s)$.*

**Proof.** Let $\xi$ be a flow in $N$ and let $\pi$ be the packing of $H$ given by $\pi(x_{out}y_{in}) = \xi(x,y)$, for $(x,y) \in A$, and $\pi(x_{in}x_{out}) = \lambda(x_{in},x_{out}) - \xi(X,x)$, for $x \in X_I$. Then

$$||\pi|| = |\xi| + 2C(X_I) - |X_I|$$

$$= g(V) + |\xi| - C(s).$$

For the converse note that the packing $\pi$ saturates all of the vertices of $H$ except possibly $t_{in}$. In fact (using essentially the same argument as in the proof of Lemma 3.2) it is easily shown that any packing $\pi$ of $H$ can be converted to a packing $\pi'$ with this property and $||\pi'|| \geq ||\pi||$. By our choice of degree bounds $g$ and $f$, we note that for every vertex $x \in X_I$ we have $deg_{\pi'}x_{in} \geq deg_{\pi'}x_{out}$. If we define $\xi'(x,y) = \pi(x_{out}y_{in})$ it is clear that $\xi'(X,x) \geq \xi'(x,X)$, for all $x \in X_I$. The edges $(x,y) \in A$ with $\xi'(x,y) > 0$ define a subgraph of $N$ that we can assume to be acyclic (otherwise the value of $\xi'$ can be reduced on some directed cycles without changing the value of $\xi'(X,t)$ until the resulting subgraph is acyclic). Thus while $\xi'$ is not necessarily a legal flow function it is easily transformed into a flow function $\xi$ with $|\xi| = ||\pi|| - g(V) + C(s)$.

$\square$

**Remark 3.14.** The transformation of Lemma 3.13 provides a linear reduction of the (integer) network flow problem to the (integer) bipartite ($g<f$)-packing problem. Thus these problems along with the (integer) bipartite ($f,f$)-packing problem and the (integer) bipartite ($g,f$)-packing problem are linearly equivalent (cf. Remark 3.12 and also Corollary 3.18).

## 3.4. Non-bipartite vs. bipartite (g,f)-packing

In the presence of general (i.e. non-integer) capacities the non-bipartite and bipartite ($g,f$)-packing problems are linearly equivalent.

Let $H = (V,E,\lambda,g,f)$ be any capacitated graph. Consider the capacitated bipartite graph $H' = (V',E',\lambda',g',f')$ constructed from $H$ as follows:

$$V' = \{\dot{v},\ddot{v} | v \in V\},$$

$$E' = \{\dot{v}\ddot{w},\dot{w}\ddot{v} | vw \in E\},$$

$$g'(\dot{v}) = g'(\ddot{v}) = g(v),$$

$$f'(\dot{v}) = f'(\ddot{v}) = f(v), \text{ and}$$

$$\lambda'(\dot{v}\ddot{w}) = \lambda'(\dot{w}\ddot{v}) = \lambda(vw).$$

**Lemma 3.15.** *$H$ admits a packing $\pi$ of size $z$ if and only if $H'$ admits a packing $\pi'$ of size 2 $z$.*

**Proof.** If $\pi$ is a packing of $H$ then the packing $\pi'$ given by $\pi'(\dot{v}\ddot{w}) = \pi'(\dot{w}\ddot{v}) = \pi(vw)$ has size $2||\pi||$. Conversely, if $\pi'$ is any packing of $H'$ then $\pi$ given by $\pi(vw) = (\pi'(\dot{v}\ddot{w}) + \pi'(\dot{w}\ddot{v}))/2$ is a packing of $H$. Furthermore, $deg_\pi v = (deg_{\pi'}\dot{v} + deg_{\pi'}\ddot{v})/2$ and so

$$h_\pi(v) = \min\{g(v),\ deg_\pi v\}$$

$$= \min\{2g(v),\ deg_{\pi'}\dot{v} + deg_{\pi'}\dot{v}\}/2$$

$$\geq (\min\{g(v), deg_{\pi'}\dot{v}\} + \min\{g(v), deg_{\pi'}\dot{v})\}/2$$

$$= (h_{\pi'}(\dot{v}) + h_{\pi'}(\dot{v}))/2.$$

Thus $||\pi|| \geq ||\pi'||/2$.

$\square$

**Remark 3.16.** In light of Lemma 3.15 and Remark 3.12, we note that the $(g,f)$-packing problem is equivalent to the network flow problem. Thus Theorem 2.1 is seen to be equivalent to the well-known max-flow min-cut theorem [14]. In fact using the reduction of Lemma 3.13 it is straightforward to derive the max-flow min-cut theorem from Theorem 2.1.

If we assume that $H$ is an integer capacitated graph and that, in addition, $g(v) < f(v)$ for all $v \in V$, then Lemma 3.15 can be strengthened to the following.

**Lemma 3.17.** *If $g(v) < f(v)$ for all $v \in V$ then $H$ admits an integer packing $\pi$ of size $z$ if and only if $H'$ admits an integer packing $\pi'$ of size $2z$.*

**Proof.** As shown in the proof of Lemma 3.15 integer packings of $H$ translate directly to integer packings of $H'$ and integer packings of $H'$ to half-integer packings of $H$. So it suffices to argue that among maximum size half-integer packings of $H$ there exists an integer packing.

Let $\pi$ be any half-integer packing of $H$ and let $\pi^*$ be a half-integer packing of $H$ of size at least $||\pi||$ that minimizes the number of non-integer edge assignments. Consider the set $J$ of edges of $H$ which are given fractional values of $\pi^*$. If $vw \in J$ then, without loss of generality, $deg_{\pi^*}v \leq g(v)$ and $deg_{\pi^*}w = f(w)$. (If both $deg_{\pi^*}v < f(v)$ and $deg_{\pi^*}w < f(w)$ then $\pi^*(vw)$ can be increased by $1/2$. If both $deg_{\pi^*}v > g(v)$ and $deg_{\pi^*}w > g(w)$ then $\pi^*(vw)$ can be decreased by $1/2$.) Hence, as $g(v) < f(v)$ for all $v \in V$, the subgraph induced by $J$ is bipartite. Furthermore, in this subgraph all vertices of odd degree must have $deg_{\pi^*}v < g(v)$. Thus, by a straightforward Eulerian path type argument, the edges of $J$ can be decomposed into a set of even cycles and even length paths joining vertices with $deg_{\pi^*}v < g(v)$. If we alternately add and subtract $1/2$ to the edges of either such configuration we construct a packing with smaller $J$, a contradiction. Hence $J = \phi$; that is $\pi'$ is integer.

$\square$

**Corollary 3.18.** *The integer $(g<f)$-packing problem and the integer bipartite $(g<f)$-packing problem are linearly equivalent.*

**Proof.** It suffices to observe that the reduction described in the proof of Lemma 3.17 can be implemented in $O(|E|)$ steps by a straightforward depth-first search. $\square$

**Remark 3.19.** It follows from Remarks 3.14 and 3.16 that the $(g,f)$-packing problem is linearly reducible to the $(g<f)$-packing problem. Furthermore, the integer bipartite $(g,f)$-packing problem is linearly reducible to the integer bipartite $(g<f)$-packing problem. In light of Corollary 3.18, we should not expect a linear reduction from the integer $(g,f)$-packing problem to the integer $(g<f)$-packing problem for non-bipartite graphs. Such a reduction would yield an algorithm for the integer $(g,f)$-packing problem that avoids the complications of blossoms.

Figure 3.1 summarizes the reductions of this section. We denote by $A \xrightarrow{i} B$ the fact that problem A is linearly reducible to problem B (using the construction in Lemma or Theorem i). Double arrows indicate that the reductions preserve integrality.

FIGURE 3.1 HERE

Using essentially the same transformations as described in Lemmas 3.15 and 3.17, Anstee [3] presents a two-phase solution of integer $(g,f)$-factor problem. His first phase solves what he calls the directed $(g,f)$-factor problem using a direct reduction to the maximum flow problem. The second phase - transforming a directed $(g,f)$-factor to an undirected $(g,f)$-factor - is a matching-type problem. As Anstee points out [3, Corollary 6.2], the second phase is trivial in the event that $g<f$. This is just another way of stating the fact that the integer $(g<f)$-packing problem is linearly reducible to the maximum flow problem.

There is another approach, like that of Anstee [3], to the integer $(g,f)$-factor problem which first solves a maximum flow problem and then a matching-type problem. If $H = (V,E,\lambda,g,f)$ is an integer capacitated graph, then consider the integer capacitated graph $H^- = (V,E,\lambda,g^-,f)$ where $g^-(v) = \min\{g(v),f(v)-1\}$, for all $v \in V$. Obviously $H^-$ is an instance of the $(g<f)$-packing problem and its solution can be found by reduction to the maximum flow problem. If $H^-$ does not admit a perfect packing then neither does $H$. Alternatively a maximum size packing of $H$ can be constructed from a perfect packing of $H^-$ by matching-type augmentations.

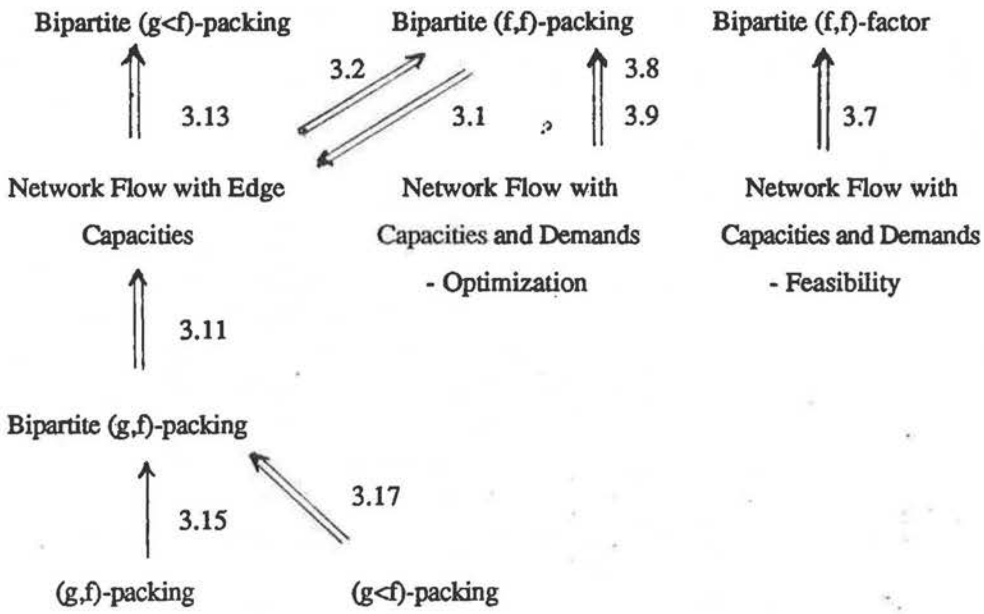


Figure 3.1. Linear reductions between packing and flow problems

# 4. ALGORITHMS FOR INTEGER (g,f)-PACKINGS

The results of the previous section, specifically Lemmas 3.11 and 3.15, combine to show that in looking for efficient algorithms for the general $(g,f)$-packing problem one need go no further than the rather extensive literature on efficient algorithms for finding maximum flows in networks. In particular, by the maximum-flow algorithm of Sleator and Tarjan (cf. [29]), we know that the general $(g,f)$-packing problem can be solved in $O(|V||E|\log|V|)$ time. Furthermore, these same results, together with Remark 3.12, demonstrate an $O(\min(|V||E|\log|V|,\sqrt{f(V)}\lambda(E)))$ upper bound on the complexity of the integer bipartite $(g,f)$-packing problem.

In this section we look more carefully at the integer $(g,f)$-packing problem. We demonstrate an $O(\sqrt{g(V)}\lambda(E))$ time solution. (In Appendix 1, we describe an alternative approach - essentially a generalization of Hopcroft and Karp's bipartite matching algorithm - that is possible when the underlying graph is bipartite). Note that, by Remark 2.3, it suffices to prove our bound for unit capacitated graphs. Hereafter all capacitated graphs will be unit capacitated (i.e. $\lambda \equiv 1$) and whenever we refer to packings it should be understood that we mean unit packings, i.e. essentially subgraphs.

Let $H = (V,E,\lambda,g,f)$ denote an arbitrary unit capacitated graph. If $\pi$ and $\rho$ are packings of $H$ then $S(\pi,\rho)$ denotes the set of edges $\{e \in E | \pi(e)+\rho(e) = 1\}$.

## 4.1. The case $g \equiv f$

First, we consider the case of $(f,f)$-packings. Here, by a direct generalization of [21] we have:

**Lemma 4.1.** *Suppose $\pi$ and $\rho$ are packings of $H$ and $||\pi|| < ||\rho||$. Then there exist $(||\rho|| - ||\pi||)/2$ edge disjoint augmenting paths in $H$ with respect to $\pi$ using edges of $S(\pi,\rho)$ only.*

**Proof.** $S(\pi,\rho)$ obviously has a partition into alternating paths. Decompose $S(\pi,\rho)$ into a minimal set $S$ of alternating paths. By minimality, no two paths start (or end) at the same vertex with edges of opposite type. Hence, if a path starts or ends at a vertex $v$ with an edge of $\rho$ then $deg_\pi v < deg_\rho v \le f(v)$. Thus every path in $S$ that starts and ends with an edge of $\rho$ must contain as a subpath an augmenting path with respect to $\pi$. But, by straightforward counting, at least $(||\rho||-||\pi||)/2$ paths in $S$ must start and end with an edge of $\rho$.

□

If $P$ is an augmenting path with respect to the packing $\pi$, we denote by $Aug(\pi;P)$ the result of augmenting $\pi$ along $P$. Similarly, if $S$ is any edge disjoint set of augmenting paths with respect to the packing $\pi$, then $Aug(\pi;S)$ denotes the result of augmenting $\pi$ along all of the paths in $S$.

□

**Lemma 4.2.** *If $\pi$ is any packing of H, P is a shortest augmenting path with respect to $\pi$ in H, and $P'$ is any augmenting path with respect to $Aug(\pi;P)$, then*

$$|P'| \geq |P| + 2|P \cap P'|.$$

**Proof.** Straightforward modification of Theorem 2 in [21].

□

As observed by Gabow [15] these lemmas imply that a packing of maximum size in $H = (V,E,\lambda,f,f)$ can be found by starting with the empty packing $\rho_0$ and applying $O(\sqrt{f(V)})$ augmentation phases each of which finds, and augments along, a maximal edge disjoint set of minimum length augmenting paths with respect to the current packing. (cf. Algorithm A below)

**Algorithm A**
$\rho \leftarrow \rho_0$
**while** $\rho$ is not a maximum packing **do begin**
/next phase/
find a maximal set $S$ of minimum length edge
disjoint augmenting paths
$\rho \leftarrow Aug(\rho;S)$ **end**
**report** $\rho$

Gabow completes his demonstration of an $O(\sqrt{f(V)}|E|)$ upper bound for the $(f,f)$-packing problem by proving that each phase can be implemented in $O(|E|)$ steps.

**Lemma 4.3.** [15] *If $\pi$ is any packing in H then a maximal edge disjoint set of minimum length augmenting paths with respect to $\pi$ can be found in $O(|E|)$ steps.*

□

Gabow's proof uses a non-trivial reduction to the problem of finding a maximal edge disjoint set of minimum length augmenting paths with respect to a matching in an associated graph (and thereafter the results of [27]). Gabow's reduction employs what he calls sparse substitutes. The reduction preserves bipartiteness and so it could, in principle, exploit the relative simplicity of Hopcroft and Karp's bipartite matching algorithm [21] in the case where $H$ is bipartite. However, it should be noted that when $H$ is bipartite Gabow's reduction can be avoided by a direct (and simple) generalization of a phase of Hopcroft and Karp's bipartite matching algorithm (see Appendix).

## 4.2. The case $g \not\equiv f$

Suppose now that $g \not\equiv f$. Gabow [15] observed that the problem of finding a $(g,f)$-factor of $H$ with the maximum number of edges can be reduced to two instances (no larger than the original) of the special case of the same problem in which $g \equiv 0$. This special case is precisely the $(f,f)$-packing problem. Thus, the results above provide an

$O(\sqrt{f(V)}|E|)$ algorithm for finding a $(g,f)$-factor of $H$ with the maximum number of edges. The first of the two subproblems addresses the question of feasibility (does there exist a $(g,f)$-factor) and the solution described by Gabow can be interpreted as an $O(\sqrt{f(V)}|E|)$ algorithm for the $(g,f)$-packing problem. The reader should recall that this is precisely the same upper bound that we get for the integer bipartite $(g,f)$-packing problem using a reduction to the maximum flow problem. (cf. Remark 3.12) As we shall show, this bound can be tightened in the case that $g(V)$ is $o(f(V))$.

We start by recalling Gabow's reduction of the $(g,f)$-packing problem to the $(f,f)$-packing problem. As before, let $H = (V,E,\lambda,g,f)$ be a unit capacitated graph. We denote by $H^*$ the unit capacitated graph $(V^*,E^*,\lambda^*,g^*,f^*)$ where

$$V^* = \{\dot{v},\ddot{v} \mid v \in V\}$$

$$E^* = \{\dot{v}\dot{w},\ddot{v}\ddot{w} \mid vw \in E\} \cup C$$

$$g^*(\dot{v}) = g^*(\ddot{v}) = f^*(\dot{v}) = f^*(\ddot{v}) = f(v), \qquad \text{for } \textit{each } v \in V,$$

and $C$, the set of cross edges of $H^*$, contains, for each $v \in V$, precisely $f(v) - g(v)$ copies of the edge $\dot{v}\ddot{v}$.

$H^*$ with the cross edges removed (and degree constraints ignored) can be viewed as the union of two copies, call then $\dot{H}$ and $\ddot{H}$, of $H$. We will sometimes refer to $\dot{H}$ and $\ddot{H}$ as the two *sides* of $H$. We say that vertices $\dot{v}$ and $\ddot{v}$ and edges $\dot{v}\dot{w}$ and $\ddot{v}\ddot{w}$ are *reflections* of one another in $H^*$. If $G$ is any subgraph of $H^*$ then the reflection of $G$, denoted $r(G)$, is the graph whose vertices (respectively edges) are the reflections of the vertices (respectively edges) of $G$.

Let $\pi$ be any packing of $H$. We denote by $\pi^*$ the packing of $H^*$ defined by $\pi^*(\dot{v}\dot{w}) = \pi^*(\ddot{v}\ddot{w}) = \pi(vw)$, for all $vw \in E$ and $\pi^*(\dot{v}\ddot{v}) = 1$ for exactly $f(v) - \max\{g(v), deg_\pi v\}$ copies of the edge $\dot{v}\ddot{v}$, for all $v \in V$. We say that a packing $\rho$ of $H^*$ is *symmetric* if $\rho = \pi^*$ for some packing $\pi$ of $H$ (and in this case we denote $\pi$ by $\hat{\rho}$).

**Property 4.4.** *(a) There exists a maximum size packing of $H^*$ which is symmetric.*
*(b) If $\pi_1$ and $\pi_2$ are any two packings of $H$, then* $||\pi_1{}^*|| - ||\pi_2{}^*|| = 2(||\pi_1|| - ||\pi_2||)$.

**Proof.** Every packing $\rho$ of $H^*$ gives rise to two symmetric packings $\rho_1$ and $\rho_2$, with $||\rho_1||+||\rho_2|| = 2||\rho||$, formed by reflecting the packing induced by $\rho$ on each of the two sides of $H^*$. Property (b) is immediate from the definitions.

□

By property 4.4 a maximum size symmetric packing $\rho$ of $H^*$ induces a maximum size packing $\hat{\rho}$ of $H$.

We wish to exploit the fact that finding a maximum packing in a capacitated graph of the form $H^*$ is not an arbitrary instance of the $(f,f)$-packing problem. As before, we will search for augmenting paths in phases where each phase identifies a maximal set of edge disjoint minimum length paths and successive phases identify paths of progressively greater length. However, we will impose the added restriction that each set of minimum length paths preserves the symmetry of the initial packing. A set of paths $H^*$ is *symmetric* if it is closed under reflection, that is the reflection of every path in the set also belongs to the set.

> **Algorithm** B
> $\rho \leftarrow \rho_0{}^*$
> **while** $\rho$ is not a maximum packing **do begin**
> /next phase/
> find a symmetric maximal set $S$ of minimum length edge disjoint augmenting paths
> $\rho \leftarrow Aug(\rho;S)$ **end**
> **report** $\hat{\rho}$

We first show that there is no loss of generality in restricting our attention to symmetric sets of disjoint augmenting paths.

**Lemma 4.5.** *If $\rho$ is a symmetric packing of $H^*$ then there exists a symmetric maximal set of edge disjoint minimum length augmenting paths with respect to $\rho$ in $H^*$.*

**Proof.** Let $S$ be a maximal symmetric set of edge disjoint minimum length augmenting paths with respect to $\rho$ in $H^*$. Suppose that $P$ is an augmenting path with respect to $\rho$, that $P$ is edge disjoint from all paths in $S$ and that $P$ has the same length as the paths in $S$. Suppose first that $P$ contains no cross edge. Then, by the symmetry of $S$, $r(P)$ is edge disjoint from all paths in $S \cup \{P\}$. Hence $S \cup \{P, r(P)\}$ is a symmetric set of edge disjoint minimum length augmenting paths with respect to $\rho$, contradicting the maximality of $S$. Alternatively, we can express $P$ as the concatenation of paths $P_1$, $P_2$ and $P_3$ where $P_1$ and $P_3$ are maximal subpaths containing no cross edge and $|P_2| \geq 1$. Suppose, without loss of generality, that $|P_1| \leq |P_3|$. Then the path $P'$ formed by connecting $P_1$ and $r(P_1)$ by the appropriate cross edge is a symmetric augmenting path with respect to $\rho$ of length $2|P_1| + 1 \leq |P|$. Since $P'$ must be disjoint from all of the paths in $S$, it follows that $S \cup \{P'\}$ is a symmetric set of edge disjoint minimum length augmenting paths with respect to $p$, again contradicting the maximality of $S$. Thus no such path $P$ can exist and $S$ must be maximal even among non-symmetric sets of edge disjoint augmenting paths with respect to $\rho$.

□

We say that a path $P$ in $H^*$ is *normal* if it uses no cross edge or if $\{P\}$ is symmetric (i.e. $P = r(P)$). Careful inspection of the proof of Lemma 4.5 reveals a somewhat stronger assertion namely, if $\rho$ is a symmetric packing of $H^*$ then there exists a symmetric maximal set of edge disjoint minimum length normal augmenting paths with respect to $\rho$ in $H^*$.

It follows from Lemma 4.5 that, assuming we start with a symmetric packing of $H^*$, we are guaranteed that the packing $\rho_i$ formed after the i-th phase of Algorithm B is symmetric. According to the following lemma our restriction to symmetric maximal sets of minimum length augmenting paths does not change the complexity of a single phase of the algorithm (cf. Lemma 4.3).

**Lemma 4.6.** *If $\pi$ is any symmetric packing in $H^*$ then a symmetric maximal edge disjoint set of minimum length normal augmenting paths with respect to $\pi$ can be found in $O(|E^*|)$ steps.*

**Proof.** We use Gabow's reduction [15] of the problem of finding a maximal edge disjoint set of minimum length augmenting paths with respect to $\pi$ in $H^*$ to the problem of finding a maximal edge disjoint set of minimum length augmenting paths with respect to a matching $M$ in an associated graph denoted $H_k{}^*$. We note that $H_k{}^*$ retains the symmetric structure $H^*$ and that $M$ retains the symmetry of $\pi$. Only minor modifications of the matching algorithm of [27] lead to an algorithm that discovers only symmetric sets of normal augmenting paths with respect to $M$ in $H_k{}^*$ (in fact, the entire search can be confined to one of the two sides of $H_k{}^*$).

□

It remains to bound the number of phases required to find a maximum size symmetric packing in $H^*$ using only symmetric sets of augmenting paths. Here (at last) we are able to turn the restriction to symmetric sets of augmenting paths to our advantage.

**Lemma 4.7.** *If $\rho$ and $\pi$ are symmetric packings of $H^*$ and $||\rho|| > ||\pi||$ then there exists a symmetric set $S$ of $(||\rho|| - ||\pi||)/2$ edge disjoint normal augmenting paths in $H^*$ with respect to $\pi$ using edges of $S(\pi,\rho)$ only.*

**Proof.** It is easy to see that the edges of $S(\pi,\rho)$ can be partitioned into a minimal symmetric collection $S'$ of edge disjoint normal alternating paths. By a simple counting argument, at least $\rho(E^*) - \pi(E^*) = (||\rho||-||\pi||)/2$ of these paths must both start and end with an edge of $\rho$. By minimality and symmetry of $S'$, if $v$ is an endpoint of a path ending with an edge of $\rho$ then it is not the endpoint of a path ending with an edge of $\pi$, and hence $deg_\pi v < deg_\rho v \le f^*(v)$. It follows that all paths in $S'$ starting and ending with an edge of $\rho$ must contain, as a subpath, a normal augmenting path with respect to $\pi$ in $H^*$. Again by the symmetry of $S'$, we can find a symmetric set $S$ of at least $(||\rho||-||\pi||)/2$ edge disjoint subpaths all of which are normal augmenting paths with respect to $\pi$ in $H^*$.

□

**Corollary 4.8.** *If $\rho$ and $\pi$ are symmetric packings of $H^*$ and $||\rho|| \ge ||\pi||$ then there exists a normal augmenting path $P$ with respect to $\pi$ in $H^*$, using edges of $S(\pi,\rho)$ only, with $|P| \le 2t+3$, where $t = \pi(E^*-C)/(||\hat{\rho}||-||\hat{\pi}||)$.*

**Proof.** Suppose all paths in $S$ have length at least $2l+3$. Then together they account for at least $l(||\rho||-||\pi||)/2$ non-cross edges of $\pi$. Since there are at most $\pi(E^*-C)$ such edges in total, it follows that

$$l \leq 2\pi(E^*-C)/(||\rho||-||\pi||)$$

$$= \pi(E^*-C)/(||\hat{\rho}||-||\hat{\pi}||),$$

by Property 4.4(b).

□

Let $\rho$ be any symmetric packing of $H^*$ and let $\rho'$ be the packing formed from $\rho$ by augmentation along the paths in some set of edge disjoint normal augmenting paths.

**Lemma 4.9.** $\rho'(E^*-C) - \rho(E^*-C) \leq (||\rho'||-||\rho||)$.

**Proof.** Each augmentation increases $||\rho||$ by 2 and $\rho(E^*-C)$ by at most 2.

□

Let $\pi_0$ denote the empty packing of $H$ (i.e. $\pi_0(e) = 0$, for all $e \in E$).

**Corollary 4.10.** *If $\rho$ is a packing of $H^*$ formed from $\pi_0{}^*$ by augmentation along paths in a sequence of symmetric sets of edge disjoint normal augmenting paths, then $\rho(E^*-C) \leq 2||\hat{\rho}||$.*

**Proof.** It suffices note that, by Property 4.4(b), $||\rho|| - ||\pi_0{}^*|| = 2(||\hat{\rho}|| - ||\pi_0||) = 2||\hat{\rho}||$.

□

**Lemma 4.11.** Algorithm B uses $O(\sqrt{||\hat{\rho}||})$ phases suffices to find a symmetric packing $\rho$ of maximum size in $H^*$.

**Proof.** Let $\rho_i$ denote the packing formed by our algorithm after the i-th phase ($\rho_0 = \pi_0{}^*$). Choose $j$ so that $||\hat{\rho}_{j-1}|| < ||\hat{\rho}||-\sqrt{||\hat{\rho}||} \leq ||\hat{\rho}_j||$.

By Corollary 4.8 the paths discovered during the j-th phase all have length at most

$$2\rho_{j-1}(E^*-C)/\sqrt{||\hat{\rho}||} + 3$$

$$\leq 4||\hat{\rho}_{j-1}||/\sqrt{||\hat{\rho}||} + 3, \qquad \text{by Corollary 4.10}$$

$$< 4\sqrt{||\hat{\rho}||}.$$

Hence, since path lengths increase with each phase, $j < 4\sqrt{||\hat{\rho}||}$. Since $||\hat{\rho}_{i+1}|| \geq ||\hat{\rho}_i|| + 1$, for all $i \geq 0$, there are less than $5\sqrt{||\hat{\rho}||}$ phases in total.

$\square$

**Theorem 4.12.** *$O(\sqrt{||\pi||}|E|)$ time suffices to construct a packing $\pi$ of maximum size in $H$.*

**Proof.** By Lemmas 4.6 and 4.11, a maximum size symmetric packing $\rho$ in $H^*$ can be constructed in $O(\sqrt{||\hat{\rho}||}|E^*|) = O(\sqrt{||\hat{\rho}||}|E|)$ time. But, by property 4.4, $\hat{\rho}$ is a maximum size packing in $H$.

$\square$

**Corollary 4.13.** *The integer $(g,f)$-packing problem can be solved in $O(\sqrt{g(V)}\lambda(E))$ time.*

**Remark.** For simplicity we have described our algorithm for $(g,f)$-packing as a variant of the first phase of Gabows edge maximum $(g,f)$-factor algorithm. This renders our original problem first into an $(f,f)$-packing problem of a special form, and then into a matching problem. If our concern were for the most efficient implementation then it is clear that the overhead of these successive reductions could be reduced by interpreting our algorithm as a search for augmenting paths in the original constrained graph.

This alternate view is particularly valuable when the original graph is bipartite. In this case, our $(g,f)$-packing algorithm can be rexpressed as a rather natural generalization of Hopcroft and Karp's bipartite matching algorithm. This means that the bipartite $(g,f)$-packing problem (and, in light of Corollary 3.17, the $(g<f)$-packing problem) both admit $O(\sqrt{g(V)}|E|)$ time algorithms that avoid the complexities of blossoming. In the appendix we describe this alternative approach in more detail.

# 5. OTHER OPTIMIZATION CRITERIA AND AN NP-COMPLETENESS RESULT

Let $H = (V,E,\lambda,g,f)$ be a unit capacitated graph. The "unit version" of the problem we have been studying, to maximize the size of a unit packing, is only one of several possible optimization problems concerned with degree-constrained subgraphs. For most of these problems the set of feasible solutions is the same, namely the set of all unit packings, or equivalently, the set of all unit $(0,f)$-factors. They differ in their choice of the objective function:

(1) maximize the size; cf. section 1

(2) minimize the deficiency, [25]; since deficiency $= g(V)$ - size, (2) is equivalent to (1).

(3) maximize the number of edges, [12]; this problem is often referred to as the b-matching problem [7,9]. Note that (3) is equivalent to the special case of (1) when $g = f$, i.e., that a unit $(0,f)$-factor with maximum number of edges is a unit $(f,f)$-packing of maximum size and vice versa.

(4) maximize the number of edges among all unit $(g,f)$-factors, [15,28]; this optimization problem has as its feasible set the set of all unit $(g,f)$-factors and hence requires, as a first step, a method to find a unit $(g,f)$-factor. Solving (1) finds a $(g,f)$-factor if one exists, and otherwise provides in some sense a unit $(0,f)$-factor closest to being a unit $(g,f)$-factor.

(5) maximize the number of saturated vertices; this is another sense in which a unit $(0,f)$-factor could be closest to a unit $(g,f)$-factor. (Recall that a vertex $v$ is saturated by $G$ if $deg_G v \geq g(v)$.)

Problems (1) - (4) all admit polynomial algorithms. The purpose of this section is to demonstrate the NP-completeness of the last variant (5); specifically, we shall prove the NP-completeness of the following decision problem:

**THE MAXIMUM SATURATION PROBLEM**

INSTANCE: A unit capacitated graph $H = (V,E,\lambda,g,f)$ and an integer $t \leq |V|$.

QUESTION: Is there a unit packing $\pi$ of $H$ saturating at least $t$ vertices?

To prove our result we shall use the k-dimensional matching problem, well known to be NP-complete when $k \geq 3$:

**THE k-DIMENSIONAL MATCHING PROBLEM**

INSTANCE: A set $P$ of $l$ $k$-tuples $p^i = <p_1^i,p_2^i,...,p_k^i>$, $i = 1,...,l$, with each $p_j^i \in \{1,2,...,n\} = Z_n$.

QUESTION: Is there a subset $S \subseteq P$ of $n$ $k$-tuples such that no two elements of $S$ agree in any coordinate?

**Lemma 5.1.** The $k$-dimensional matching problem is polynomially reducible to the maximum saturation problem.

**Proof.** We assume that $k \geq 2$. Let $P = \{<p_1^i,p_2^i,...,p_k^i| 1 \leq i \leq l, p_j^i \in Z_n\}$ be an instance of the $k$-dimensional matching problem. Let $h$ be any integer, $h \geq k$. We shall transform $P$ into an instance of the maximum saturation problem. We shall use many

copies of one simple graph $M$ illustrated in Figure 5.1; to distinguish amongst the copies we shall use subscripts. Thus $M_\alpha$ is the unit capacitated graph $(V_\alpha, E_\alpha, \lambda, g, f)$ where

$$V_\alpha = \{q_\alpha, r^1_\alpha, \ldots, r^{h-1}_\alpha, s^1_\alpha, \ldots, s^{k-1}_\alpha, t_\alpha\},$$

$$E_\alpha = \{(q_\alpha r^i_\alpha), (r^i_\alpha s^j_\alpha), (s^j_\alpha t_\alpha) | 1 \le i \le h\text{-}1, 1 \le j \le k\text{-}1\}, \text{ and}$$

$$g(v) = k \quad \text{and} \quad f(v) = h, \quad \text{for all } v \in V_\alpha.$$

FIGURE 5.1 HERE

Let $\pi$ be any packing of $M_\alpha$. Since $t_\alpha$ has degree $k-1$ in $M_\alpha$ it can not possibly be saturated. Furthermore, since vertex $r^i_\alpha$ has degree $k$, it can be saturated only if all of its incident edges $e$ have $\pi(e) = 1$. Thus at most $deg_\pi q_\alpha$ such vertices are saturated. Hence $\pi$ saturates at most $k + deg_\pi q_\alpha$ vertices in total.

Now let $H_P$ denote the unit capacitated graph formed by taking $nk$ disjoint copies of $M_\alpha$, $\alpha = <1,1>, \ldots, <n,k>$, and adding the vertices $p^1, \ldots, p^l$ and the edges $\{(p^i q_\alpha | \alpha = <j, p^i_j>, 1 \le i \le l, 1 \le j \le k\}$, with $g(p^i) = k$ and $f(p^i) = h$, $1 \le i \le l$. Since $H_P$ can be constructed from $P$ in polynomial time it suffices to show that $P$ admits a $k$-dimensional matching if and only if $H^P$ admits a unit packing saturating at least $n + nk(h + k - 1)$ vertices.

Let $\{p^{i_1}, \ldots, p^{i_n}\}$ be a $k$-dimensional matching of $P$. Consider the packing $\pi$ of $H_P$ defined by $\pi(p^i, q_{<d,p^i_d>}) = 1$, for $i \in \{i_1, \ldots, i_n\}$ and $1 \le d \le k$, $\pi(e) = 1$, for $e \in E_\alpha$ $\alpha \in \{<1,1>, \ldots, <n,k>\}$, and $\pi(e) = 0$ elsewhere. It is easy to confirm that $\pi$ saturates exactly $n + nk(h + k - 1)$ vertices.

Conversely, suppose that $H_P$ admits a packing $\pi$ saturating at least $n + nk(h + k - 1)$ vertices. It follows that $\pi$ saturates some number $m \ge n$ of the vertices $p^1, \ldots, p^l$. Let $d = \sum_{e \in E^*} \pi(e)$, where $E^*$ is the set of edges incident with the vertices $\{p^1, \ldots, p^l\}$. Then $d \ge mk$. Partition $E^*$ into $\bigcup_\alpha E_\alpha{}^*$, where $E_\alpha{}^*$ consists of those edges of $E^*$ that are incident with $q_\alpha$, and let $x_\alpha = \sum_{e \in E_\alpha{}^*} \pi(e)$. Then $\sum_\alpha x_\alpha = d$ and $\sum_\alpha (x_\alpha - 1) = d - nk$. By the observation above, the number of saturated vertices in $M_\alpha$ is at most $k + h - 1 - (x_\alpha - 1)$ and so the total number of saturated vertices is at most

$$m + nk(h+k-1) - (d-k_n) = n + nk(h+k-1) - (k-1)(m-n) - (d-km).$$

It follows that $m = n$, $d = mk$, each vertex $q_\alpha$ is incident with exactly one edge $e = (p^i, q_\alpha)$ for which $\pi(e) \ne 0$, and for that edge $e$, $\pi(e) = 1$. Thus the saturated vertices among $p^1, \ldots, p^l$ represent a $k$-dimensional matching of $P$.

$\square$

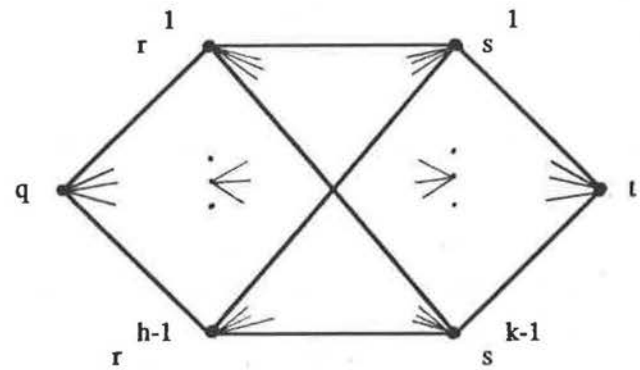


Figure 5.1. The graph M.

The $(g,f)$-factor problem in which $g \equiv a$ and $f \equiv b$ for fixed integers $a$ and $b$ is sometimes referred to as the *$[a,b]$-factor problem.*

**Corollary 5.2.** *The maximum saturation problem is NP-complete even for $[a,b]$-factors, provided $a \geq 3$ for all $v \in V$.*

**Proof.** This follows immediately from the construction above and the fact that the $k$-dimensional matching problem is $NP$-complete for $k \geq 3$.

$\square$

**Remark 5.3.** If $g \equiv 0$ the maximum saturation problem is trivial (since all vertices are automatically saturated) and if $g \equiv 1$ then a polynomial solution follows from the observation that the maximum number of saturated vertices is precisely the size of the maximum packing. The complexity of the only remaining case, when $g \equiv 2$, remains an open question, even in the case of $[2,b]$-factors.

$\square$

**Remark 5.4.** Problems (1)-(4) relaxed to general (not necessarily unit or integer) packings also admit polynomial solutions; for example, each can be formulated as a linear program of size polynomial in the size of the problem. However the maximum saturation problem (5) remains $NP$-complete also in the case when we only seek a (general) packing saturating at least $t$ vertices. In fact, we have been careful to write the proof of Lemma 5.1 in such a way that it also applies to this situation.

# APPENDIX: Finding maximum packings in bipartite graphs

In this appendix we consider the problem of efficiently constructing packings of maximum size in integer capacitated bipartite graphs. As we noted in Remark 2.3, it suffices to consider the case of unit capacitated bipartite graphs. Throughout this section $B = (V_L \cup V_R, E, \lambda, g, f)$ denotes an arbitrary unit capacitated bipartite graph and $\pi_0$ denotes the trivial packing of $B$ (i.e. $\pi_0(e)=0$, for all $e \in E$).

## A.1. Algorithm overview

### A.1.1. Monotonicity of augmenting path lengths

Following Hopcroft and Karp [21] we would like to show that if we use the strategy of augmenting along shortest available paths then the length of successive augmenting paths increases monotonically. Unfortunately, this is not the case in this new setting. Because of the more complex nature of our augmenting paths, specifically the variety of path endings, it is possible for a short path to be created as a result of an augmentation along a longer path. In the case of bipartite graphs this non-monotonicity, though still present, is easily circumvented.

### A.1.2. Two phase algorithm

The high level structure of our bipartite packing algorithm differs from the bipartite matching algorithm of Hopcroft and Karp [21] in one important respect. Since augmenting paths for matchings in bipartite graphs must start and end at deficient vertices in different sides of the graph, it is sufficient to search for augmenting paths starting from only one of the two sides. In the case of our more general packings augmenting paths may start and end on the same side and may end at non-deficient vertices. The first approach that suggests itself is to look simultaneously for augmenting paths with starting points in either side. For reasons similar to those discussed above, this forces us to either accept non-monotonicity in the lengths of augmenting paths or adopt an unnatural definition of path length. It turns out to be more straightforward to search for paths in two phases each of which restricts attention to starting points in a single side. (It remains to be demonstrated, of course, that such an approach will not overlook any paths.)

Our algorithm takes the following form:

**Algorithm** ***left–right–maximum***

$\pi \leftarrow \pi_0$
{phase A: find and augment along augmenting paths with starting points in $V_L$}
    $\pi \leftarrow$ *left–maximum* $(\pi, H)$
{phase B: find and augment along augmenting paths with starting points in $V_R$}
    $\pi \leftarrow$ *right–maximum* $(\pi, H)$
**report** $\pi$

### A.1.3. Algorithm correctness

Note that the two phases of algorithm ***left-right-maximum*** are structurally identical - the only difference being the source of starting points for augmenting paths. With the following lemma we can focus our attention (as far as questions of correctness and efficiency are concerned) on the implementation of a single phase.

We say that an augmenting path that starts in $V_L$ (respectively $V_R$) is a *left* (respective *right)* augmenting path. If $B$ admits no left (respectively right) augmenting path with respect to the packing $\pi$ then we say that $\pi$ is a *left* (respectively *right)-maximum* packing in $B$. Obviously, $\pi$ is maximum if and only if it is both left- and right-maximum.

**Lemma A.1.** *Let $\pi$ be any left-maximum packing in $B$ and let $P$ be any right augmenting path with respect to $\pi$. Then $\pi' = Aug(\pi;P)$ is a left-maximum packing in $B$.*

**Proof.** Consider the following recursive assignment of colours to the vertices of $B$.

i) if $v \in V$ and $deg_\pi v < g(v)$, colour $v$ *black*;

ii) if $v$ is black and $\pi(vw) = 0$, colour $v$ *white*; and

iii) if $v$ is white and $\pi(vw) = 1$, colour $v$ *black*.

We claim that no vertex is coloured both black and white. Suppose to the contrary that $v \in V$ is assigned both colours. Then there exist alternating paths to v starting at deficient vertices in both $V_L$ and $V_R$. The union of two such paths must be an alternating path joining deficient vertices in $V_L$ and $V_R$. But the latter must contain an augmenting path starting in $V_L$, contradicting the left-maximality of $\pi$.

It follows that $P$ must be vertex-disjoint from every alternating path starting from a deficient vertex in $V_L$. Hence the augmentation of $\pi$ along $P$ does not change the set of alternating paths starting from deficient vertices in $V_L$. In particular, the resulting packing in left-maximum.

$\square$

**Corollary A.2.** *Assuming the phases A and B are correctly implemented the algorithm* left-right-maximum *finds a packing of maximum size in $B$.*

$\square$

## A.2. The existence of short augmenting paths

Let $\pi$ be a unit packing of $B$. We define $||\pi||_L$ to be $\sum_{v \in V_L} h_\pi(v)$. We start by proving a somewhat stronger version of Corollary 2.5 for unit capacitated bipartite graphs. Let $\pi$ and $\rho$ be any two unit packings of $B$. Define

$$D(\pi,\rho) = \{v \in V_L | h_\rho(v) > h_\pi(v)\} \text{ and } S(\pi,\rho) = \{vw \in E | \pi(vw) \neq \rho(vw)\}.$$

The packing $\pi \dot{-} \rho$ is defined by

$$(\pi \dot{-} \rho)(e) = \max\{0, \pi(e) - \rho(e)\}.$$

The packing $\rho \dot{-} \pi$ is defined similarly.

**Lemma A.3.** *If $||\rho||_L > ||\pi||_L$ then there exists a left augmenting path with respect to $\pi$ in $B$, starting at a vertex in $D(\pi,\rho)$, using edges of $S(\pi,\rho)$ only, and having length at most $2t+2$, where $t = ||\pi||_L/(||\rho||_L - ||\pi||_L)$.*

**Proof.** Suppose that $\pi$ and $\rho$ provide a counterexample that minimizes $t = ||\pi||_L/(||\rho||_L - ||\pi||_L)$. Let $D = D(\pi,\rho)$, $N = \{v | \rho(uv) > \pi(uv)$, for some $u \in D\}$, and $M = \{w | \pi(vw) > \rho(vw)$, for some $v \in N\}$. Clearly, $deg_\pi v < g(v)$, for all $v \in D$. Moreover, we can assume that $deg_\pi v = f(v)$ for all $v \in N$ and $deg_\pi w \leq g(w)$ for all $w \in M$, since otherwise there is an obvious augmenting path of length at most 2 starting at a vertex in $D$. It follows that

$$||\rho||_L = ||\pi||_L + \sum_{u \in D} (h_\rho(u) - h_\pi(u))$$

$$\leq ||\pi||_L + \sum_{u \in D} deg_{\rho \dot{-} \pi} u$$

$$\leq ||\pi||_L + \sum_{v \in N} deg_{\rho \dot{-} \pi} v$$

$$\leq ||\pi||_L + \sum_{v \in N} deg_{\pi \dot{-} \rho} v, \quad \text{since } deg_\pi v = f(v) \geq deg_\rho v, \text{ for } v \in N$$

$$\leq ||\pi||_L + \sum_{w \in M} deg_\pi w$$

$$\leq 2||\pi||_L, \text{ since } deg_\pi w \leq g(w), \text{ for } w \in M, \text{ and } M \subseteq V_L.$$

Hence we can assume that $t \geq 1$.

Define $\pi'$ and $\rho'$ to be the following packings of $B$.

$$\pi'(vw) = \begin{cases} 0 & \text{if } v \in N,\ vw \in E, \text{ and } \pi(vw) > \rho(vw) \\ \pi(vw) & \textit{otherwise} \end{cases}$$

$$\rho'(vw) = \begin{cases} 0 & \text{if } v \in N,\ vw \in E, \text{ and } \pi(vw) < \rho(vw) \\ \rho(vw) & \textit{otherwise.} \end{cases}$$

Then, since
(*) $h_\rho(u) - deg_\rho u \le h_{\rho'}(u) - deg_{\rho'} u$, for $u \in D$,
(**) $deg_\pi w \le g(w)$, for $w \in M$, and
(***) $deg_\pi v \ge deg_\rho v$ and $deg_{\pi'} v = deg_{\rho'} v$, for $v \in N$,
it follows that

$$\begin{aligned} \|\rho\|_L - \|\rho'\|_L &= \sum_{u \in D} (h_\rho(u) - h_{\rho'}(u)) \\ &\le \sum_{u \in D} (deg_\rho u - deg_{\rho'} u), \quad \textit{by } (*) \\ &\le \sum_{v \in N} (deg_\rho v - deg_{\rho'} v) \\ &\le \sum_{v \in N} (deg_\pi v - deg_{\pi'} v), \quad \textit{by } (***) \\ &= \sum_{w \in M} (deg_\pi w - deg_{\pi'} w) = \sum_{w \in M} (h_\pi(w) - h_{\pi'}(w)), \quad \textit{by } (**) \\ &= \|\pi\|_L - \|\pi'\|_L. \end{aligned}$$

Thus $\|\pi'\|_L \le \|\pi\|_L + \|\rho'\|_L - \|\rho\|_L$. Furthermore, since $h_{\rho'}(v) \le h_{\pi'}(v) \le h_\pi(v)$, for $v \in D$, and $h_{\rho'}(v) \le h_\rho(v) \le h_\pi(v)$ elsewhere, it follows that $\|\rho'\|_L \le \|\pi\|_L$. Thus

$$\|\pi'\|_L / (\|\rho'\|_L - \|\pi'\|_L) \le \|\pi\|_L / (\|\rho\|_L - \|\pi\|_L) - 1 = t - 1.$$

By the supposed minimality of the counterexample, it follows that $B$ admits a left augmenting path $P'$ with respect to $\pi'$, starting at a vertex $w \in D(\pi', \rho') \subseteq M$, using edges of $S(\pi', \rho') \subseteq S(\pi, \rho)$, and having length at most $2t$. It is easy to confirm that $P'$ uses no vertex of $D \cup N$. Hence $P'$ can be extended to a left augmenting path $P$ with respect to $\pi$, of length at most $t + 2$, by adding, as a prefix, a pair of edges $uv$ and $vw$ where $u \in D$, $v \in N$, $\pi(uv) < \rho(uv)$ and $\pi(vw) > \rho(vw)$. This contradicts our hypothesis and establishes the lemma.

□

## A.3. Algorithms for left-maximum packing

We proceed to describe algorithms that implement phase A of algorithm *left-right-maximum,* that is find a left-maximum packing by a sequence of augmentations along left augmenting paths. (Algorithms for phase B follow by replacing left by right.) Our strategy is identical to that of the most efficient known algorithms for bipartite matching [21] and maximum flow [10,23,26]. Augmenting paths are discovered by depth-first search in a (breadth-first) labelled subgraph of the host graph $B$.

### A.3.1. Assignment of vertex labels

Given a unit packing $\pi$ of $B$ we define a *breadth-first labelling* $l = l_\pi$ of $V = V_L \cup V_R$ as follows:

i) Initially $l(v) = 0$ for all vertices $v \in V_L$ with $deg_\pi v < g(v)$; all other vertices are unlabelled and all vertices are unscanned.

ii) In general, having scanned vertices labelled $<i$, we proceed to scan each vertex $v$ for which $l(v) = i$ as follows:

    if $i$ is odd
        then for each unlabelled $w$ with $\pi(vw) = 1$
            do $l(w) \leftarrow i + 1$
        else for each unlabelled $w$ with $\pi(vw) = 0$
            do $l(w) \leftarrow i + 1$

iii) Repeat ii) until every labelled vertex has been scanned.

Let $V_i = \{v \in V | l_\pi(v) = i\}$. The *depth* of the packing $\pi$, denoted $d_\pi$, is the smallest integer $d$ satisfying one of,

i) d is odd and for some $w \in V_d$, $deg_\pi w < f(w)$; or

ii) d is even and for some $w \in V_d$, $deg_\pi w > g(w)$.

If no such integer exists then we define $d_\pi = \infty$.

It is an immediate consequence of the proceeding definitions that

$$deg_\pi v \begin{cases} < g(v) & \text{if } l_\pi(v) = 0 \\ = f(v) & \text{if } 0 < l_\pi(v) < d_\pi \text{ and } l_\pi(v) \textit{ is odd} \\ = g(v) & \text{if } 0 < l_\pi(v) < d_\pi \text{and } l_\pi(v) \textit{ is even.} \end{cases}$$

It should be clear that a straightforward breadth-first implementation of the labelling procedure will[5], in time $O(|E|)$, compute the labels $l_\pi(v)$, the depth $d_\pi$, the vertex sets $V_i$, $0 \leq i \leq d_\pi$, and the edge sets

$$F[v] = \begin{cases} \{w | l_\pi(w) = l_\pi(v) + 1 \text{ and } \pi(vw) > 0\} & \text{if } l_\pi(v) \textit{ is odd} \\ \{w | l_\pi(w) = l_\pi(v) + 1 \text{ and } \pi(vw) = 0\} & \text{if } l_\pi(v) \textit{ is even} \end{cases}$$

[5] We assume that $|V|$ is $O(|E|)$. This is guaranteed if, for example, $B$ has no isolated vertices.

for $v$ satisfying $0 \leq l_\pi(v) < d_\pi$.

The vertices $\bigcup_{0 \leq i \leq d_\pi} V_i$, together with the edges $vw$, $v \in \bigcup_{0 \leq i < d_\pi} V_i$, $w \in F[v]$, form a layered subgraph of $B$ that we refer to as the *core* of $B$ (with respect to the packing $\pi$). It is clear that $d_\pi$ is finite if and only if $B$ admits a left augmenting path with respect to $\pi$ using edges of the core only. According to the following Lemma, the finiteness of $d_\pi$ is a necessary and sufficient condition for the existence of *any* left augmenting path in $B$ with respect to $\pi$ (equivalently, the search for such augmenting paths may be restricted to the core of $B$).

**Lemma A.4.** *Let $\pi$ be an packing of $B$. Then $B$ admits a left augmenting path with respect to $\pi$ if and only if $d_\pi$ is finite.*

**Proof.** Sufficiency is clear from the definitions of depth and augmenting path. To prove necessity suppose that $B$ admits a left augmenting path with respect to $\pi$. Let $P = v_0, v_1, \ldots, v_k$ be any such path of minimum length. Since $P$ is an alternating path, it follows from the labelling procedure that $l_\pi(v_i) \leq i$ and $l_\pi(v_i) \equiv i \bmod 2$, for $0 \leq i \leq k$. In fact $l_\pi(v_i) = i$, for $0 \leq i \leq k$, since otherwise $P$ could be shortened by replacing $v_0, v_1, \ldots, v_i$ with an alternating path of length $l_\pi(v_i)$. Thus $d_\pi = k$.

$\square$

**Corollary A.5.** *A packing $\pi$ is left-maximum if and only if $d_\pi = \infty$.*

$\square$

### A.3.2. High level description

Lemma A.5 (together with its proof) suggests that a search for left augmenting paths with respect to a packing $\pi$ in $B$ can be restricted to what we call $l$-paths, where $l$ denotes some labelling of $B$. An *$l$-path* with respect to $\pi$ is an augmenting path $P = v_0, v_1, \ldots, v_k$ in which $l(v_i) = i$, $0 \leq i \leq k$. Note that the lemma suggests the choice $l = l_\pi$, but we will need to consider other possible labellings as well.

If $P_1, P_2, \ldots, P_t$ is a sequence of paths such that $P_i$ is an $l$-path in $B$ with respect to $\pi_{i-1}$, where $\pi_0 = \pi$ and $\pi_i = Aug(\pi_{i-1}; P_i)$, for $i \geq 1$, then we refer to the sequence as an *$l$-batch* in $B$ with respect to $\pi$. In addition we denote $\pi_i$ by $Aug(\pi; P_1, P_2, \ldots, P_i)$, for $i \geq 1$. An $l$-batch is said to be *maximal* if it cannot be extended to a larger $l$-batch. By the nature of augmentations, it should be clear that all of the paths in an $l$-batch are edge-disjoint.

It is an immediate consequence of Corollary A.6 that the following straightforward incremental algorithm constructs a left-maximum packing of $B$ given an initial packing $\pi_0$.

**Algorithm** *incremental left-max*

$\pi \leftarrow \pi_0$
assign $l_\pi$ labels to $V$
**while** $d_\pi < \infty$ **do begin**
    $P \leftarrow$ *an* $l_\pi$*-path in* $B$
    $\pi \leftarrow Aug(\pi;P)$
    *assign* $l_\pi$ *labels to* $V$ **end**
**report** $\pi$

In fact, the algorithm is easily seen to run in $O(g(V_L)|E|)$ steps since the execution of the loop body is dominated by the cost of the labelling procedure and every augmentation increases the size of the packing by at least 1.

A more efficient algorithm for left-maximum packings is based on the idea (like that of Dinic [10] and successors [23,26] in connection with the maximum flow problem, and Hopcroft and Karp [21] and successors [27] in connection with the maximum matching problem) of finding augmenting paths in batches. Specifically, the $l_\pi$-paths identified in *incremental left-max* can be discovered in a sequence of phases where all of the paths discovered in any one phase are $l_\pi$-paths with respect to the *same* $\pi$ (i.e., an $l_\pi$-batch).

**Algorithm** *batched left-max*

$\pi \leftarrow \pi_0$
assign $l = l_\pi$ labels to $V$
**while** $d_\pi < \infty$ **do begin**
    **while** *there exist l-paths in B with respect to* $\pi$ **do begin**
        $P \leftarrow$ *an l-path in* $B$
        $\pi \leftarrow Aug(\pi;P)$ **end**
    *assign* $l = l_\pi$ *labels to* $V$ **end**
**report** $\pi$

### A.3.3. Algorithm analysis

The analysis of *batched left-max* is based on two considerations namely the worst case behaviour of the body of the outer loop (i.e. the cost of finding an $l$-batch) and the number of repetitions of this loop. Since all paths in a batch are edge-disjoint a straightforward depth-first search (essentially identical to that used in Hopcroft and Karp's bipartite matching algorithm [21]) can be used to identify (and augment along) the paths in a maximal batch in $O(|E|)$ steps. In the remainder of this section we develop an upper bound on the number of batches required to achieve a left-maximum packing. As in the maximum flow [10] and maximum matching [21] problems, the key observation is that the length of augmenting paths in successive batches is strictly increasing.

**Lemma A.6.** *Let* $\pi$ *be an packing of* $B$, $l = l_\pi$ and $d_\pi < \infty$. Let $P_1,P_2,\ldots,P_t$ be an l-batch, $\pi' = Aug(\pi;P_1,\ldots,P_t)$, and $l' = l_{\pi'}$. Then,

i) $deg_\pi v = deg_{\pi'} v$, when $0 < l(v) < d_\pi$; and

ii) $l'(v) \geq l(v)$, when $0 \leq l'(v) \leq d_\pi$

**Proof.** Assertion i) is an immediate consequence of the fact that each $P_i$ is an $l$-path and augmentations along $l$-paths do not change the degree (with respect to the current packing) of any of their interior vertices.

To prove ii) suppose, to the contrary, that for some vertex $v$, $l'(v) \leq d_\pi$ and $l'(v) < l(v)$. Among all such vertices choose $v$ so that $l'(v)$ is minimized. Clearly, $l'(v) > 0$ since augmentations never create new unsaturated vertices. Suppose that $l'(v)$ was assigned a value when vertex $w$ was being scanned. Hence $l'(w) = l'(v) - 1$ and, by our minimality assumption, $l'(w) \geq l(w)$. Now either the edge $wv$ belongs to some path $P_i$ or $\pi'(wv) = \pi(wv)$. In either case we note that $l(v) \leq l(w) + 1 \leq l'(w) + 1 = l'(v)$, contradicting our assumption.

□

**Lemma A.7.** *Let $\pi$ be a packing of $B$ with $d_\pi < \infty$. Let $P_1,P_2,...,P_k$ be an $l_\pi$-batch in $B$ with respect to $\pi$ and $\pi' = Aug(\pi;P_1,...,P_k)$. Then $d_{\pi'} \geq d_\pi$. Furthermore, if $P_1,...,P_k$ is a maximal $l_\pi$-batch then $d_{\pi'} > d_\pi$.*

**Proof.** Let $l = l_\pi$, $l' = l_{\pi'}$, $d = d_\pi$, and $d' = d_{\pi'}$. If $d' = \infty$ then there is nothing to prove. Let $P = v_0,...,v_{d'}$ be any $l'$-path with respect to $\pi'$.

Suppose that $d' < d$. Then, by Lemma A.7, we have $deg_\pi v_{d'} = deg_{\pi'} v_{d'}$, and $l(v_{d'}) \leq l'(v_{d'})$. Since $l(v_{d'}) \equiv l'(v_{d'})$ (mod 2), it follows immediately from the definition of depth that $d \leq l(v_{d'}) = d'$, a contradiction.

Suppose now that $d' = d$. Then, by Lemma A.7, we know that $l(v_i) \leq l'(v_i)$, for $0 \leq i \leq d$, and $deg_\pi v_i = deg_{\pi'} v_i$, provided $l(v_i) < d$. Suppose that $l(v_i) < l'(v_i)$, for some $i < d$. Then, since $l(v_i) \equiv l'(v_i)$ (mod 2) for $0 \leq i \leq d$, it follows from the labelling procedure that $l(v_{i+1}) \leq l(v_i) + 1 \leq l'(v_i) - 1 < l'(v_{i+1})$. It follows, by induction on $i$, that $d \leq l(v_{d'}) < l'(v_{d'}) = d'$, contradicting our hypothesis. Thus, $l(v_i) = l'(v_i)$, for $0 \leq i \leq d$, that is $P$ is an $l$-path with respect to $\pi$. Hence $P_1,P_2,...,P_k$ is not a maximal $l$-batch.

□

**Corollary A.8.** *If $\pi$ is a packing of $B$ formed after $i$ iterations of the outer loop of procedure batched left-max* then $d_\pi > i$.

□

**Lemma A.9.** *Algorithm* batched *left-max constructs a left-maximum packing $\pi$ in $B$ in $O(|E|\sqrt{||\pi||})$ time.*

**Proof.** It suffices to show that, given any initial packing $\pi_0$, algorithm *batched left-max* constructs a left-maximum packing $\pi$ in $B$ using $O(\sqrt{||\pi||})$ batches (i.e. iterations of the outer loop).

Suppose that the size of the packing constructed by *batched left-max* exceeds $||\pi|| - \sqrt{||\pi||}$ for the first time following the $i$-th batch ($i = 0$ if $||\pi_0||$ exceeds this bound). By Corollary A.2, the $l$-paths discovered during the $i$-th batch all have length at most $2(||\pi|| - \sqrt{||\pi||})/\sqrt{||\pi||} + 2 \leq 2\sqrt{||\pi||}$. Hence, by Corollary A.9, $i \leq 2\sqrt{||\pi||}$. Since each subsequent batch increases the size of the current packing by at least 1, there are at most $3\sqrt{||\pi||}$ batches in total.

□

## A.4. The complexity of integer bipartite $(g,f)$-packing

Noting that $||\pi|| \leq g(V)$ for any packing $\pi$ of B, it follows from the results of the preceeding section that Algorithm *left-right-maximum* can be implemented to run in worst cast time $O(|E|\sqrt{g(V)})$. We summarize this in the following theorem.

**Theorem A.10.** *A maximum packing of an unit capacitated bipartite graph $B = (V,E,\lambda,g,f)$ (equivalently, a solution to the unit bipartite $(g,f)$-packing problem) can be found in worst-case time $O(|E|\sqrt{g(V)})$.*

□

**Corollary A.11.** *A solution to the unit $(g<f)$-packing problem can be found in worst-case time $O(|E|\sqrt{g(V)})$.*

**Proof.** Immediate from the Theorem and Corollary 3.17.

□

In light of Remark 2.3, the results of Theorem A.10 and Corollary A.11 can be stated more generally as,

**Corollary A.12.** The integer bipartite $(g,f)$-packing problem and the integers $(g<f)$-packing problem can both be solved in worst-case time $O(\lambda(E)\sqrt{g(V)})$.

**Remark A.13.** Recalling Remark 3.14, we note that Corollary A.12 provides another proof of Theorem 3.4, thus allowing a network problem to be solved in time $O(\sqrt{C(X)}a(A))$ via our bipartite packing algorithm. We do not know if a network flow algorithm of complexity $O(\sqrt{C(X)}c(A))$ (such as [10]) can be used to solve the integer bipartite $(g,f)$-packing problem in time $O(\lambda(E)\sqrt{g(V)})$.

---

# REFERENCES


[1] J. Akiyama, D. Avis, H. Era, On a {1,2}-factor of a graph, TRU Math 16 (1980), 97-102.

[2] A. Amahashi, M. Kano, On factors with given components, Discrete Math. 42 (1982), 7-26.

[3] R. Anstee, An algorithmic proof of Tutte's f-factor theorem, J. Algorithms 6 (1985), 112-131.

[4] H.B. Belck, Regulare Factoren von Graphen, J. Reine Angew. Math., 188 (1950), 228-252.

[5] C. Berge, Two theorems in graph theory, Proc. Nat. Acad. Sc., USA 43 (1957), pp. 842-844.

[6] C. Berge, Sur le couplage maximum d'un graphe, C.R. Acad. Sciences Paris 247 (1958), pp. 258-259.

[7] C. Berge, *Graphs and Hypergraphs,* North Holland, Amsterdam, 1973.

[8] C. Berge, M. Las Vergnas, On the existence of subgraphs with degree constraints, Proc. Nederl. Akad. Wetensch, Amsterdam, series A, volume 81, 2 (1978), pp. 165-176.

[9] G. Cornuejols, W. Pulleyblank, A matching problem with side conditions, Discrete Math. 29 (1980), 135-159.

[10] E.A. Dinic, Algorithm for solution of a problem of maximum flow in a network, with power estimation, Soviet Math. Dokl. 11 (1970), 1277-1280.

[11] J. Edmonds, Paths, trees, and flowers, Canad. J. Math. 17 (1965), 449-467.

[12] J. Edmonds, E.L. Johnson, Matching: a well-solved class of integer linar programs, in *Combinatorial Structures and Their Applications,* (R.K. Guy et al., eds.), Gordon and Breach, N.Y. 1970, pp. 89-92.

[13] S. Even, R.E. Tarjan, Network flow and testing graph connectivity, SIAM J. Comput. 4(1975), 507-518.

[14] L.R. Ford, Jr., D.R. Fulkerson, *Flows on Networks,* Princeton University Press, Princeton, N.J., 1962.

[15] H.N. Gabow, An efficient reduction technique for degree constrained subgraphs and bidirected network flow problems, Proc. 15th Annual ACM Symposium on Theory of Computing (1983), pp. 448-456.

[16] T. Gallai, On factorization of graphs, Acta Math. Hung 1 (1950), 133-153.

[17] P. Hell, D.G. Kirkpatrick, Star factors and star packings, TR 82-6, Dept. of Computing Science, Simon Fraser University, Burnaby, British Columbia, 1982.

[18] P. Hell, D.G. Kirkpatrick, Packings by complete bipartite graphs, SIAM J. Alg. Disc. Math. 7 (1986), 199-209.

[19] P. Hell, D.G. Kirkpatrick, B. Li, Rounding in graphs and symmetric matrices, to appear.

[20] K. Heinrich, P. Hell, D.G. Kirkpatrick, G.Z. Liu, A simple characterization for $(g<f)$-factors, to appear.

[21] J.E. Hopcroft, R.M. Karp, An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs, SIAM J. Comput. 2 (1973), 225-231.

[22] M. Kano, A. Saito, A short proof of Lovasz's factor theorem, Akashi Institute of Technology TR 26 (Jan. 1984), 167-170.

[23] A.V. Karzanov, Determining the maximal flow in a network by the method of preflows, Soviet Math. Dokl. 15 (1974), 434-437.

[24] M. Las Vergnas, An extension of Tutte's 1-factor theorem, Discrete Math. 23 (1978), 241-255.

[25] L. Lovasz, Subgraphs with prescribed valencies, J. Combinatorial Th. 8 (1970), 391-416.

[26] V.M. Malhotra, M. Pramodh Kumar, S.N. Maheshwari, An $O(|V|^3)$ algorithm for finding maximum flows in networks, Inf. Proc. Letters 7, 6 (October 1978), 277-278.

[27] S. Micali, V.V. Vazirani, An $O(\sqrt{|V|}\cdot|E|)$ algorithm for finding maximum matchings in general graphs, Proc. 21st Annual Symposium on Foundations of Computer Science (1980), pp. 17-27.

[28] Y. Shiloach, Another look at the degree constrained subgraph problem, Information Processing Lett. 12 (1981), 9-92.

[29] R.E. Tarjan, Data Structures and Network Algorithms, SIAM CBMS-NSF Regional Conference Series in Applied Mathematics, 1983. *Network Algorithms.*

[30] W.T. Tutte, The factorization of linear graphs, J. London Math. Soc. 22 (1947), 107-111.

[31] W.T. Tutte, The factors of graphs, Canad. J. Math. 4 (1952), 314-328.

[32] W.T. Tutte, Graph factors, Combinatorica 1 (1981), 79-97.

[33] W.T. Tutte, The method of alternating paths, Combinatorica 2, 3(1982), 325-332.