NOTES ON INTEGER BIN-PACKING FOR TECHNOLOGY MAPPING ON TREES

MICHAEL D. HUTTON

INTEGER-BIN-PACKING: Given a list L of n items of integer size/weight, and a integer bin-capacity K, pack the items into a minimum number of bins.

The decision version of this problem is NP-Complete [4]. The well-known heuristic "first-fit-decreasing" (FFD) is known to approximate the minimum bin-packing within a factor of $\frac{11}{9}$ [4]. The FFD algorithm simply sorts the elements in decreasing order, and applies a greedy algorithm: the current item is placed in the first bin in which it fits. If K is a constant, the problem is polynomial time by exhaustive search.

For a technology mapping application[2, 3], Francis [2] recently showed that FFD provides the optimum bin-packing strategy when $K = 5$ or 6, and furthermore when $K = 5$ the FFD packing has the property that its largest *hole* is maximum among all other optimum packings.

We can generalize this result if we view the packing problem as a matching problem; i.e. we construct a graph G whose nodes are the items of L, and items x and y share an edge if the sum of their values is no larger than the bin-capacity (i.e. $w(x) + w(y) \leq K$). If G arises from some set of items in this way we say G is a *pairing graph*. If the bin-capacity is such that no more than two items can ever fit in one bin (excluding items of size 1, which we dispose of shortly), we claim that the FFD bin-packing is guaranteed to have both the minimum number of bins, and the largest hole property mentioned above; i.e. it induces a maximum matching with the largest hole property. The optimality with respect to size is alluded to by exercise 10.16 of [5]; however, the hole size property is interesting and necessary for the application, and is not mentioned.

First we want to exclude instances with items of size 1 from consideration; we will call these items *trivial items*:

LEMMA 0.1. *If instance (L, K) of INT-BINPACK provides a counter-example to the optimality (in terms of number of bins) of the FFD algorithm when there are trivial items (weight 1) in L, then the instance (L', K) where L' is L with all trivial items removed also provides a counter-example to the optimality of FFD.*

Proof. FFD adds 1's only after all other elements have been placed, and hence introduces a new bin only if all other bins are entirely full. Thus it cannot make an optimum packing non-optimum by adding trivial items. ∎

We henceforth assume that there are no trivial items in L. Note that 1's could still have an effect on the hole property, but we will defer this issue.

1. FFD and the Matching Matroid. Define the set system $(V, \mathcal{M})$ on the vertices of G, where $A \in \mathcal{M}$ iff there is a matching M of G whose vertices *cover* the set A. Clearly if $A \in \mathcal{M}$ and $A' \subseteq A$ then $A' \in \mathcal{M}$ also, so we say that $\mathcal{M}$ has the *inclusion property* on its members.

LEMMA 1.1. *If $A_p, A_{p+1} \in \mathcal{M}$, where $|A_p| = p$ and $|A_{p+1}| = p + 1$, then there exists some $x \in A_{p+1} - A_p$ such that $A_p + x \in \mathcal{M}$—i.e. some x in A_{p+1} can augment A_p to larger cardinality in $\mathcal{M}$.*

Notation: here we use $X - Y$ to denote the set difference of X and Y: the set of all elements in X but not in Y, and $|X|$ to denote the cardinality of set X.

Proof. Fix minimal cardinality matchings M_p and M_{p+1} which cover A_p and A_{p+1} respectively. (By minimality, each edge must have at least one endpoint in the covered set.) Let $x \in A_{p+1} - A_p$, and let y be its *mate* in M_{p+1}. If $y \notin A_p$ then $M_p + xy$ is a matching of G and $A_p + x \in \mathcal{M}$; so assume otherwise.

Now consider the maximal path P_x in G which begins with the edge $xy = a_0a_1$, alternates between M_{p+1} edges and M_p edges, and ends with the first vertex a_k which is outside of $A_p \cap A_{p+1}$. If $a_{k-1}a_k$ is an M_{p+1} edge ($a_k \notin A_p$) then P_x is an augmenting path—applying it to M_p gives a matching M_p' which covers $A_p + x + a_k$, so $A_p + x \in \mathcal{M}$. Otherwise $a_0 = x \in A_{p+1} - A_p$ and $a_k \in A_p - A_{p+1}$, so choose a different x and continue. Since $|A_{p+1}| > |A_p|$, $|A_{p+1} - A_p| > |A_p - A_{p+1}|$, so at least one such x in $A_{p+1} - A_p$ gives an augmenting path P_x as in the previous case, so $A_p + x \in \mathcal{M}$ for that x. ∎

COROLLARY 1.2. *If A, A' are maximal elements of $\mathcal{M}$ then $|A| = |A'|$.*

Proof. Otherwise the smaller set can be augmented by some element of the larger. ∎

COROLLARY 1.3. *$(V, \mathcal{M})$ is a matroid.*

Proof. Lemma 1.1 and Corollary 1.2 are equivalent definitions of a matroid. ∎

Define now the node-weighted maximum matching (NWMM) problem as follows: for a graph $G = (V, E)$ and integer weighting function $w(v)$ on V, find a matching M such that the sum of node weights of the vertices covered by M is a maximum. As mentioned previously, [5] suggests the existence of the aforementioned matroid as a proof that the "greedy algorithm" solves the NWMM problem. The matching matroid was originally defined in [1]. The "greedy algorithm" referred to considers the vertices one at a time by decreasing weight: at stage i, the algorithm adds vertex x_i to the current set A if $A + x_i$ is covered by some matching in G, otherwise x_i is discarded.

COROLLARY 1.4. *The greedy algorithm correctly solves the NWMM problem.*

Proof. $(V, \mathcal{M})$ is a matroid. ∎

This does not help us immediately with our initial problem—the FFD algorithm essentially chooses edges, not nodes, and purports to run correctly in linear time (given an initially sorted list L). The greedy algorithm alluded to above for $(V, \mathcal{M})$ includes a step which tests whether there is a matching M which covers $A + x$, for which no constant time algorithm is known in general. However, the class of graphs that we are dealing with has several exploitable properties which will allow us to maintain the FFD matching M and use it to test whether $A + x$ is covered by some matching M^+ in constant time.

Consider the following augmentation to FFD, which we will call Algorithm *greedy-match*:

Algorithm greedy-match:

```
M, A ← ∅, initially all x ∈ V are unmarked.
for each x ∈ V, taken in decreasing order by w(x) do
    if x is marked then
        {Claim 1: there is some edge zx already in M.}
        A ← A + x
    elseif all vertices adjacent to x are already marked then
        {Claim 2: A + x ∉ M; i.e. it contains a matroid circuit.}
        do nothing
    else
        {Claim 3: all z > x such that zx ∈ E are already matched with some z' > x.}
```

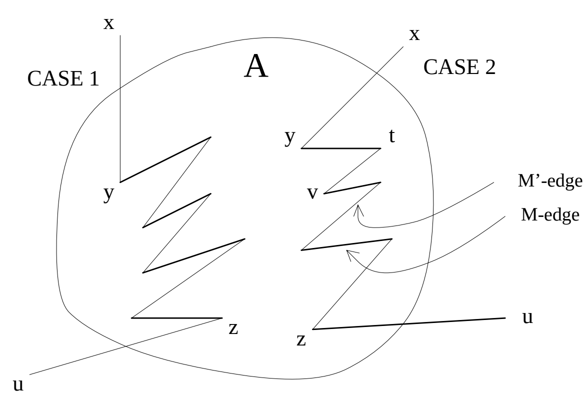
$y \leftarrow$ largest unmarked y such that $xy \in E$
mark x and y; add x to A and xy to M
{Claim 4: all $y' > y$ such that $xy' \in E$ are already matched with some $z > x$.}
end if
end for

Note that "$x > y$" denotes x precedes y in L; if we are referring to the weight of an item x we will use $w(x)$.

The four claims in the algorithm are easy to show, except possibly for Claim 2 which follows as a corollary of the next lemma. The algorithm clearly can be implemented to run in $O(|V| + |E|)$ time after sorting of V (which is also linear whenever $w(x) = O(|V|)$ for all $x \in V$).

LEMMA 1.5. *Let G be a pairing graph. At stage i of greedy-match, if any matching M' covers $A + x$, then some matching which includes all the edges of M also does; i.e. M is extendable.*

Proof. If any M' covers $A+x$ then its edges form a disjoint union of alternating paths and alternating cycles (in G) with the edges of M. One path P begins with edge $xy \in M'$, and ends with the first edge to leave A. (The connected piece, of G induced on $M + M'$, which contains x must be a path rather than a cycle as M does not cover x already.) Choose M' so that the resulting P is of minimal length. We show that if M' exists, there must also be a u not currently covered by M such that $xu \in E$, so xu augments M to be a matching covering $A + x$. If y is not already outside of A ($u = y$) we have two cases to consider (see diagram).



In case 1, P ends with a vertex u not currently covered by M. Then $u \notin A$ also. But notice that $w(x)+w(y) \leq K$ as $xy \in E$, and $w(u) \leq w(x) \leq w(y)$ by the ordering of V, so $w(x) + w(u) \leq K$ and xu is an edge which augments M to cover $A + x$.

In case 2, P ends with the edge uz—$z \in A$ matched to u with an edge of M, but $u \notin A$. Now, $w(t) + w(v) \leq K$ as tv is an edge, and $w(x) \leq w(t)$ by the ordering. Hence xv and the remainder of P forms a smaller alternating path, contradicting the minimality of P (i.e. the matching M'' which swaps xy and tv in M' for yt and xv is a matching which yields a smaller path P) so case 2 does not occur. Note that P cannot contain only 2 edges ($P = xyu$), as greedy-match would not match y with u when x, preceding u in L, is unmatched. ∎

COROLLARY 1.6. *If no such u exists, then $A + x \notin \mathcal{M}$ (equivalent to Claim 2).*

COROLLARY 1.7. *Algorithm greedy-match correctly finds a maximum node-weighted matching in pairing graphs in linear time.*

The following property of greedy-match will be useful later:

Lemma 1.8. *If M is the output of greedy-match on the pairing graph G and xy is an edge of M, then greedy-match yields $M - xy$ on the graph $G - x - y$.*

Proof. Suppose not. Let M^* be the output of greedy match on $G - x - y$. Let u be the first vertex of L which has a different mate in M^* than in $M - xy$. If u is mated to $v > u$ in M^* then v preceded u in L and had a different mate—contradiction; so u is mated to $v < u$ in M^* but to some $t \neq v$ in $M - xy$. t must be unmarked at the point where greedy-match chose uv (or some earlier vertex had a mate different from $M - xy$), so $v > t$. But then greedy-match could not have mated u with t (v was unmarked since no vertex preceding it was matched to it) in G—contradiction. ∎

2. Greedy-match and Largest Hole-Size.. The second property that Francis [2] discussed with respect to the FFD algorithm was the "largest-hole" property; he showed that FFD yielded the largest possible largest-hole among all optimum bin-packings of integer weighted L whenever $K = 5$. We need to generalize this to the case where the bin-capacity K can hold no more than two items; i.e. when K is less than 3 times the smallest item weight. For $K > 5$ this will not hold in the case of trivial items (1's), but a minor adjustment to the algorithm will allow 1's to be added without losing the largest-hole property; namely that when adding 1's, we switch from the greedy-match algorithm to adding to the current smallest-hole bin.

We noticed in the previous section that greedy match generates a matching which has maximum possible node-weight. This leads to the following observation:

Proposition 2.1. *If pairing graph G has a perfect matching save for one leftover node, then the weight of this leftover node is minimal among all maximum cardinality matchings of G.*

Proof. The node weight of the matched items is maximized by greedy-match (Corollary 1.7), so the weight of the unmatched item is minimized. ∎

This fact will allow us to prove first that greedy-match gives the largest largest-hole when pairing graph G has a perfect matching, then the same result for G a general pairing graph. In these proofs we often use the following property of pairing graphs, which follows directly from the definition of their edge set being that the sum of vertex weights is no more than K.

Property 1. *If G is a pairing graph and $x > y > z$ (in the ordering L) then the edge xy implies the edge xz, and the edge xz implies the edge yz.*

Lemma 2.2. *If a pairing graph G has a perfect matching then the (perfect matching) M of greedy-match has the largest hole property among all perfect matchings of G.*

Proof. Suppose not, and let G be a vertex-minimal counter example—a pairing graph for which some M' has a larger largest-hole than M.

Looking first at M: Consider the last vertex y added and its mate x in G. Notice that greedy-match gives $M - xy$ as a maximum matching on $G - y$ with x left over and the singleton x is a small as possible by Lemma 2.1. Define

$$T = \{t_i \in V \mid t_i < x \text{ and } xt_i \in E\}$$

and

$$Z = \{z_i \in V \mid z_i t_i \in M \text{ and } t_i \in T\},$$

so T is the set of later vertices (in L) adjacent to x, and Z is the set of vertices to which they are matched in M. Let $s = |Z| = |T|$.

Notice $z_i > x$ for all i—otherwise the edge xt_i would have been selected over $z_i t_i$ by greedy-match. Similarly $z_i x$ and $z_i z_j$ are not edges of G for any i and j or these would also be edges of M. Furthermore, if $z_i t$ is an edge then xt is also an edge, so $t \in T$; we conclude that the vertices of Z are adjacent *only* to vertices of T in $G - y$.

Now turn to M'—the hole-optimal perfect matching on G: Let the largest hole be at the edge (a, b) where $a > b$, and let v be the vertex adjacent to y in M'. (Note $v > x$ or M' contradicts Lemma 2.1.)

We have $w(b) \geq w(y)$ as y is the least vertex, and $w(a) + w(b) < w(x) + w(y)$ by our assumption on hole-size, so $w(x) > w(a) \geq w(b) \geq w(y)$. If $w(b) = w(y)$ then $M' - ab + vb$ is a matching on $G - y$ with a smaller leftover vertex than x, so, in fact, $w(x) > w(a) \geq w(b) > w(y)$.

Now $a, b \notin Z$ as $x > a, b$. If $a \in T$ then $b \in T$ also so M' leaves only $s - 2$ vertices of T along with y to match with the s vertices of Z and cannot be perfect, so $a \notin T$. Since xa cannot then be an edge, and $x > a$, then is no ux edge in G for $u > x$ so M' must match x to some $t_i \in T$. There are $s - 1$ remaining t_i's and y; these must be matched with the s vertices of Z. This means that $Z + x + T + y$ is covered by a perfect sub-matching in both M and M', which does not contain the largest-hole edge (a, b). Since Lemma 1.8 says that greedy-match acts the same on $G^* = G$ with all these vertices removed, G^* is a smaller counter-example; contradiction. ■

LEMMA 2.3. *Let G be a pairing graph. Then M provided by greedy-match has the largest-hole property among all maximum matchings of G.*

Proof. The proof is by induction on $|V|$.

The basis is trivial. Suppose M of greedy-match has the largest hole property for all pairing graphs on $< k$ vertices, and consider adding y to G. Without loss of generality assume that y is the last vertex added, since the induction hypothesis is over all G of size $< k$.

If the addition of y doesn't change M (i.e. y is added as a singleton) we can't do any better—y is the smallest weight element, so it has the largest possible hole size. Otherwise if adding y makes M perfect (covers $V(G)$ totally) then it has the largest hole property by Lemma 2.2. So assume y is matched with some singleton x by greedy-match, but some alternate matching $M' + zy$ has a larger hole than $M + xy$, and $M + xy$ and $M' + zy$ each leave one or more singletons in G. (Note that unless y is made a singleton, we can only decrease the largest-hole of a matching by adding it.) Now, greedy-match put y with x, and x must have been a singleton with the largest hole (since we have lost the largest-hole property in M by adding y). In fact, x must also have been the *only* singleton of its weight, or we would still have the same hole-size as before. Let h and h' be the (new) largest hole-sizes of $M + xy$ and $M' + zy$ respectively. Consider v of weight $\min(h', w(y))$, which thus follows y in L. We can add v to the largest hole of M', increasing the cardinality of M', but no singleton of M is adjacent to y—no singleton of M can be adjacent to a vertex of weight $w(y)$, else we would have mated it with y instead of x or x would not have had the largest hole of $M - xy$. No v of weight h' could be adjacent to a singleton of M, as all holes of M are of strictly smaller size. This contradicts the fact that greedy-match finds the optimum cardinality matching without backtracking on edges of M (Lemma 1.5). ■

REFERENCES

[1] EDMONDS, J. and D. R. FULKERSON, "Transversals and Matroid Partition," *J. Res. NBS, 69B* (1965), 147-153.

[2] FRANCIS, R. J., Ph.D. Thesis, University of Toronto, 1992.

[3] FRANCIS, R. J., J. ROSE and Z. VRANESIC, "Technology Mapping of Lookup Table-Based FPGAs for Performance," *Proc. ICCAD-91,*, Nov, 1991.

[4] GAREY, M. R. and D. S. JOHNSON, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Co., New York, 1979.

[5] PAPADIMITRIOU, C. S. and K. STEIGLITZ, *Combinatorial Optimization: Algorithms and Complexity,* Prentice-Hall, 1982.