

Date: August 17, 2016

Let a path in the edge-path-partition be called *light* if its weight is less than λ_2 , and *heavy* otherwise. We handle light and heavy paths differently. For a light path, we keep track of the total weight in the path. For lights paths, if the total weight of vertices in P is at most λ_1 , then no value induced by P is in contention for being λ^* . However, there may be many light paths whose weights are greater than λ_1 , and thus their weights, and the weights of certain of their subpaths, could be candidates for being λ^* . These are then not included in the selection list \mathcal{P} , and we shall show how to quickly resolve them in a separate selection list, \mathcal{L} .

We next discuss the representation of a heavy path. Each heavy path P is represented as a sequence of overlapping subpaths, each of weight at least λ_2 . Each vertex of P is in at most two overlapping subpaths. Each overlapping subpath, except the first, overlaps the previous overlapping subpath on vertices of total weight at least λ_2 , and each overlapping subpath, except the last, overlaps the following overlapping subpath on vertices of total weight at least λ_2 . Thus any sequence of vertices of weight at most λ_2 that is contained in the path is contained in one of its overlapping subpaths. For each overlapping subpath, we shall maintain the corresponding sorted matrix $M(P)$ (as described in Section 2), where P is the overlapping subpath excluding the top vertex of the overlapping subpath.

Initially, no path is heavy, since initially $\lambda_2 = \infty$. A heavy path P can be created in one of two ways.

1. Path P was light until the value of λ_2 was reduced by a feasibility test.
2. Path P is formed by the concatenation of two or more paths following the resolution of a problematic vertex.

If a heavy path P is created in the first way, then represent P by two overlapping subpaths that are both copies of P , arbitrarily designating one as the first overlapping subpath and the other as the second. For each overlapping subpath, create a corresponding sorted matrix. If P is the concatenation of paths, all of which were light, then do the same thing.

Otherwise, path P is formed by the concatenation of paths, with at least one of them being heavy. Do the following to generate the representation for P . While there is a light path P' to be concatenated to a heavy path P'' , combine the two as follows. If P' precedes P'' , then extend the first overlapping subpath of P'' to incorporate P' . If P' follows P'' , then extend the last overlapping subpath of P'' . This completes the description of the concatenation of light with heavy paths. While there is more than one heavy path, concatenate adjacent heavy paths P' and P'' as follows. Assume that P' precedes P'' . Combine the last overlapping subpath of P' with the first of P'' . Note that a vertex in a sorted matrix can be changed at most twice before it is in overlapping subpaths that are neither the first nor the last.

The algorithm will maintain three selection lists, \mathcal{P} , \mathcal{L} , and \mathcal{V} . The first selection list is for paths, the second selection list is for sequences of light paths formed by the resolution of a problematic vertex, and the third selection list is for handling problematic vertices whose leaf paths have been resolved.

The algorithm will employ two procedures. In procedure *handle_active_paths*, the algorithm selects the weighted and unweighted medians from \mathcal{P} and \mathcal{L} separately, tests for feasibility, and adjusts λ_1 and λ_2 accordingly. During the feasibility testing, the algorithm cleans and glues adjacent subpaths that have been resolved. When a leaf path has been completely resolved, the algorithm represents the leaf path by a single vertex with the remaining accumulated weight, $accum_wgt(v)$, as described in Section 2.

The second procedure is *handle_leaves*. For any problematic vertex with a resolved leaf path hanging off it, the procedure inserts into \mathcal{V} the weight of that vertex plus the accumulated remaining weight from the resolved leaf path. The procedure then selects the median from \mathcal{V} , performs the feasibility test, and adjusts λ_1 and λ_2 accordingly. That is, for the max-min problem, if the number of cuts is at least k , then for each vertex whose weight plus the accumulated remaining weight from the resolved leaf path is at most the median, it merges the vertex with the accumulated remaining weight from the resolved leaf path. On the other hand, if the number of cuts is less than k , then for each vertex

whose weight plus the accumulated remaining weight from the resolved leaf path is at least the median, it cuts above the parent vertex, noting that any future feasibility tests would do the same. Note that any extra vertices we had artificially included to round the total number of vertices in each path to a power of two have weight zero, which will have already been resolved by this point. If a problematic vertex is resolved, leaving a sequence of light paths, insert the sum of the light paths into \mathcal{L} .

Lemma 0.1 *Following i iterations of weighted and unweighted selection and feasibility testing separately for \mathcal{L} and \mathcal{P} , where $2^{5j} = (3/4) * (24/23)^i$, at most $1/2^{5k}$ of the subpaths, of length 2^{j-k} , in \mathcal{P} can be unresolved.*

Proof We first consider the weights assigned to submatrices in \mathcal{M} (recall that we insert submatrices into \mathcal{M} representing various subpaths with weight a function of the length of the subpath). Corresponding to subpaths of lengths $1, 2, 4, \dots, n$, procedure *mats_for_path* creates n submatrices of size 1×1 , $n/2$ submatrices of size 1×1 , $n/4$ submatrices of size 2×2 , and so on, up through one submatrix of size $n/2 \times n/2$. The total weight of the submatrices corresponding to each path length is $4n^5, n^5, n^5/4, \dots, 4n^3$, resp. It follows that the total weight for submatrices of all path lengths is less than $(4/3) * 4n^5$.

Let $wgt(M)$ be the weight assigned to submatrix M . Define the *effective weight*, denoted $eff_wgt(M)$, of a submatrix M in \mathcal{M} to be $wgt(M)$ if both its smallest value and largest value are contained in the interval (λ_1, λ_2) and $(3/4)wgt(M)$ if only one of its smallest value and largest value is contained in the interval (λ_1, λ_2) . Let $eff_wgt(\mathcal{M})$ be the total effective weight of all submatrices in \mathcal{M} . When a feasibility test renders a value (the smallest or largest) from M no longer in the interval (λ_1, λ_2) , we argue that $eff_wgt(M)$ is reduced by at least $wgt(M)/4$ because of that value.

If both values were contained in the interval (λ_1, λ_2) , and one is no longer is, then clearly it is true. If only one value was contained in the interval (λ_1, λ_2) , and it no longer is, then M is replaced by four submatrices of effective weights $wgt(M)/8 + wgt(M)/8 + (3/4)wgt(M)/8 + (3/4)wgt(M)/8 < wgt(M)/2$, so that there is a reduction in weight by

greater than $wgt(M)/4$. If both values were contained in the interval (λ_1, λ_2) , and both are no longer in, then we consider first one and then the other.

Thus every element that was in (λ_1, λ_2) but no longer is causes a decrease in $eff_wgt(\mathcal{M})$ by an amount at least equal to its weight in \mathcal{P} . Values with more than half of the total weight in R find themselves no longer in (λ_1, λ_2) .

Of a submatrix M which does have representative values contained in \mathcal{P} , then M has either two values in \mathcal{P} at a total of $2w(M)/4 = (1/2)w(M)$ or one value at a weight of $w(M)/4 = (1/3)(3w(M)/4)$. For each selection and feasibility test from \mathcal{P} and \mathcal{L} , either at least half of \mathcal{M} is contained in \mathcal{L} or contained in \mathcal{P} .

Suppose at least half of \mathcal{M} is contained in \mathcal{L} . Then following a selection and feasibility test from \mathcal{L} , at least one quarter of the values and weight from \mathcal{M} will be resolved, either determined to be smaller than an updated λ_1 , or larger than an updated λ_2 and inserted into \mathcal{P} as a part of a heavy path. Thus, following a round of selection and feasibility testing from \mathcal{P} , at least $1/8$ of \mathcal{M} will be resolved.

On the other hand, if at least half of \mathcal{M} is contained in \mathcal{P} , then a round of selection and feasibility testing from \mathcal{P} resolves at least $1/4$ of \mathcal{M} .

Thus $eff_wgt(\mathcal{M})$ decreases by a factor of at least $(1/8)(1/3) = 1/24$ per iteration.

Corresponding to the subpaths of length 2^j , there are $n/2^j$ submatrices created by *mats_for_path*. If such a matrix is quartered repeatedly until 1×1 submatrices result, each such submatrix will have weight $n^4/2^{4j-2}$. Thus when as little as $(n/2^j) * (n^4/2^{4j-2})$ weight remains, all subpaths of length 2^j can still be unresolved. This can be as late as iteration i , where i satisfies $(4/3) * 4n^5 * (23/24)^i = n^5/2^{5j-2}$, or $2^{5j} = (3/4) * (24/23)^i$. While all subpaths of length 2^j can still be unresolved on this iteration, at most $(1/2 * 1/2^4)^k = 1/2^{5k}$ of the subpaths of length 2^{j-k} can be unresolved for $k = 1, \dots, j$. Also, by Lemma 3.1, each subpath can be searched by *FTEST1* in amortized time proportional to the logarithm of its length. Thus the time to search path P on iteration i is at worst proportional to $(j/2^j)n(1 + 1/2^5 + 1/2^{10} + \dots)$, which is $O((j/2^j)n)$. From the relationship of i and j , $2^j = (3/4)^{1/5}(24/23)^{i/5}$, and $j = (1/5)\log(3/4) + (i/5)\log(24/23)$. The lemma then follows. ■