Global Journal of Pure and Applied Mathematics.
ISSN 0973-1768 Volume 14, Number 5 (2018), pp. 755–774

http://www.ripublication.com/gjpam.htm

Edge Matching with Edge Insert

Waldemar Friesen

Technical University Vienna,
D-60528 Frankfurt, Im Mainfeld 7 / 5 OG 1, Germany.

Abstract

In this paper, for a graph $G = (V, E)$ with integral edge weights greater than zero, the task of finding an vertex set M with the smallest possible weight is solved such that every vertex of V is a vertex of an edge in M. First of all, usual inputs are converted into a representation, where you can quickly access required values such as neighboring vertices, adjacent edges, edge weights and the like. Then a new edge is always inserted into an already matched edge set according to ascending weight. It is shown that one can improve a matching by changing the membership of M in appropriate paths, by alternating an edge from M and not from M, when it reduces the weight. For this, the theorem is proved that there is always such a path, if one has not yet reached the smallest weight. If one cannot find such a path, the weight is the smallest. If the old edge set is increased by a new edge with all associated vertices, then a single smallest path of the M-edge exchange is sufficient from each of the maximum two new vertices to produce the lowest weight of the new edge set. The whole thing goes essentially to the insertion of new vertices. The success occurs for all edges at the latest after $C\ (\log_2 |V|)(|V| - 1)|E|$ steps.

AMS subject classification: 68W40, Computer Science, Algorithms, Analysis of Algorithms.
Keywords: Polynomial versus Nondeterministic Polynomial, Vertex Matching.

Contents

1. Preparatory Steps

Given a graph $G = (V, E)$ with a positive edge weight function $g(e) > 0, e \in E$. We seek for a smallest matching M, that is, a subset of the edges E such that every vertex v from$V$ belongs to at least one edge of $v \in e \in M$. The three usual inputs of a graph consist in the adjacency matrix A, where $A_{vw} = 1$, if $\{v, w\}$ is in E, 0 otherwise, or only the upper triangular matrix of an adjacency matrix where all the entries A_{vw} with $w < v$ are omitted, since the edge $\{w, v\}$ is equal to the edge $\{w, v\}$. The third option is a vertex edge list. In a field, the vertices are from 1 to n, and the list of the neighbor vertices belongs to each vertex. The whole essentially corresponds to the edges. This includes the weight of each edge at the field of the vertex of the corresponding vertex list. For an edge that does not exist, that is $A_{vw} = 0$, I choose a weight, which does not occur otherwise, may be -1 or mathematically ∞.

The first two inputs are not well suited for fast access to the entries, so I convert them to a vertex edge list as follows. I first set a field of n vertices and then go to the adjacency matrix with the n^2, with the upper triangle matrix $n(n-1)/2$ entries, and insert the adjacent vertices along with the edge weights as a list. For the upper triangle matrix I have to use the values for $\{w, v\}, v < w$, for (wv) as well. I can best w connect to the edge list of v at the same time v to the edge list of w at the back. Here I put a pointer from the list entry of w at v on the list entry v at w and vice versa. For the adjacency matrix, I also enter the values for (vw) and (wv) at the same time. For the last (wv), which is mirrored at the main diagonal, and represents the opposite of the edge direction, I only check whether its matrix values coincide with the entry and weight previously written with (vw). If not the input contains an error since the adjacency matrix of a graph with non-directed edges must be symmetrical, also with regard to the weights.

If I go through the adjacency matrix, I need as many steps as the input length $= n^2$ with the upper triangular matrix with the application of the mirrored edges $2\ inputlength = 2n(n-1)/2 < n^2$, so that the vertex edge list with $2|E|$ is always less than or equal to the doubling of the first two input lengths. So I have created in a maximum of 2 inputs many steps a vertex edge list, which I will use from then on only. With the vertex edge list I can access each vertex directly in just one step. If I discover a vertex without edges, there is no vertex matching, and I break the process after a maximum of $2|E| + n$ steps. I'll take the case out.

In the course of the process, it will be seen that the edges do not represent self-contained entities. I only need them to be able to walk from one vertex to its neighbors, to go through paths, and as bearers of weight. In most cases a matching M is given whose weight is to be reduced. Only interchange paths play a role. These consist in edge sequences which do not visit an edge twice, with an M-edge following a non-M-edge and vice versa. Along these paths M-edges and non-M-edges are interchanged so that the matching property is maintained. Since an inner vertex of the path before and after the interchange is always matched by an M-edge before or behind it, difficulties only occur at the beginning or end edge of the path. If there is no othe edger from the outside of the path, the start and end edges may not be part of the path because the start or end vertex would not be matched if the single M-edge was removed. I can always

start or end with a non-M-edge. The paths I've just described are called allowed paths. A commutation path, which reduces the weight of the M-edges, I call reduction path. (I have proved the reduction theorem before in the not published article *"Integral Edge Matching with Vertex Insert"*.)

Reduction theorem. *The weight of a matching M becomes less than or equal to that of a given cover W by interchange along possibly several suitable reduction paths, M goes over to M with $g(M) \leq g(W)$. If I use a smallest matching for W, M becomes a smallest.*

Proof. The fact is illuminated, for I choose an arbitrary smaller matching W. If $g(M) \leq g(W)$ already I take the empty path which does not change anything. Otherwise I consider the failures $e \in M, e \notin W$. I take e out of M and, if an end vertex of e is not matched by M, it must be in any case matched by an edge d of W. Then I insert the corresponding edge $d \notin M, d \in W$ into M. M and W then match the one to three edges so they do not need to be changed because of the general assumption $e \in M, e \notin W$. I thus get paths from one to three edges. However if an M-edge is adjacent to a non-M-edge one can combine several paths to a larger one. Two edges with the initial requirement $e \in M, e \notin W$, $f \notin M$, $f \in W$ may be connected by a non-M-edge $d \notin M, d \in W$, ie form a path edf of length 3. Then the edge d might be inserted twice into M. If I would avoid this, exclude d from the path with f and run it first, the vertex $d \cap f$ might not be matched. In this case I need to continue the path from the maximum three edges over the edge f so that I get a path of length three to five. This afterwards can be repeated over several times to an even longer path. If several f_i lead to d, it would be enough, if I continue the path with only one, maybe f_k. Then the f_i form a star with the center $d \cap f_i$. This center point always remains matched when removing an f_i. For either $f_i, i \neq k$, is taken out before f_k, then it is matched by f_k, or f_i is removed after f_k, then it is matched by inserting d when removing f_k. So I can do the M-edge conversions in any order. The paths work on completely different edges and can therefore been interchanged as desired. In the end, all defects $e \in M, e \notin W$ are eliminated, from $e \in M$ follows $e \in W$ or $M \subseteq W$ and with this $g(M) \leq g(W)$. Then there must have been paths that have reduced the weight. Because of the independent order of the path conversions, I can run them first with same result. Then I leave the non-diminishing behind them still away, so that the weight is at most smaller. I thus attain a weight no greater than $g(W)$ by means of diminishing paths. This only proves that they exist. Finding comes later. ■

I do not go as with the proof of *"Integral Edge Matching with Vertex Insert"* from a smallest matching of all vertices with few edges whereby I then insert at each step only one new edge. This time, I always add a new edge with all the associated vertices to a smallest matched subset of the vertices starting with a single edge. In this case, it is advantageous if the edges are taken up according to increasing weight. So I arrange them in ascending order. For this purpose a method *merge insertion* is suitable, which arranges two successive entries of a list by comparison, then inserts by means of the comparison of the size of the list starts the smallest of two successive lists of 2 in a list of 4, generally merges two successive 2^i-lists by comparing the head entries to an ordered

2^{i+1}-list. You have to go up to $\lceil log_2|E| \rceil$ passes through successive lists, each time $|E|$ entries are involved. So everything happens after the step number of

$$(log_2|E|+1)|E| \leq \left(log_2 \left(\frac{n(n-1)}{2} \right) + 1 \right) |E| < (2log_2 n - 1 + 1)|E| =$$

$$2(log_2 n)|E| < 2n|E|, \;\; 2 \leq n.$$

For $1 \leq |E|$ there are at least 2 vertices, ie $2 \leq n$. The inequality $log_2 n < n$ follows for $1 \leq n$ from $n < 2^n$ with the initial assumption $1 < 2^1$ and the induction conclusion $n + 1 < 2^n + 2^n = 2^{n+1}$. As a starting list of the edges I can take the vertex edge list with the order (vw), that is, after a vertex v all its neighbors w come first in ascending order and then a smaller v before a larger one. After I have arranged the list of all the edges, represented by the adjacent vertices of a vertex in its edge list, I take the result and link the vertices of the edge list additionally forward and backward in a duplicated list of all edges.

2. Restrictions with Weight Reduction Paths

I'm not going to add the new edges to an edge set that already contains all the vertices, as in the essay *"Integral Edge Matching with Vertex Insert"*, but start with a single edge and its two vertices and insert it into a least matched vertex set, that already contains all associated vertices, always a new edge $e = \{v, w\}$ with the maximum two new vertices and all edges leading from them to an old vertex. This new vertex and edge set will then be minimized with a single smallest reduction path from a new v or new w. For this reason, I first consider which paths from a new vertex, which is added to a previously smallest matched edge set D, can cause a reduction. I only look at paths with regard to reduction. It is always possible to break a path with an edge after which the weight will no longer be lowered. I call this path short. I only consider short paths. With these paths, I do not distinguish between polylines and polylines with circles at the end or pure circles. With *"polylines"* I mean double-point-free paths, which do not return to an already visited vertex, which could be called in the German *"Streckenzuege"*. If a circle with even many edges, shortly even circle, occurs in the old edge set D, without adding the new vertices v, w, the circle starts with an M-edge and ends with a non-M-edge or vice versa. The M-edge interchange along the circle pushes the M-edges only one further and works completely on the circle without outer edge. It can, therefore, always be carried through and lowers, since D is considered to be the least matched, not the weight. Even circles alone or as part of a reduction path are thus excluded. For an odd circle, there are two non-M-edges or M-edges at the beginning. Since in the first case, two non-M-edges at the initial vertex require an additional outer M-edge for matching their common vertex, the transformation may always be performed within the circle, and it also separates out. However such a non-weight-reducing circle may be at the end of a polyline with an M-edge at the end, which then becomes a non-M-edge. The circle then serves to match the last polyline vertex. For a weight reduction, the weight increase

of the circle must be smaller than the decrease at the end of the polyline. An odd circle with two M-edges at the beginning or an additional M-edge there is out of the question. However it can hang at the end of a polyline that ends with a non-M-edge, which then becomes an M-edge and matches the beginning of the circle.

I always exclude that a path visits an edge twice, because it would be converted twice and the old condition would remain. This would usually split a path into two single paths. If a path revisits a vertex, a circle is created. Since an M-edge and a non-M-edge always follow one another in the path, two M-edges, ie double edges, must be located in the intersection vertex. In the direction of a double edge, I can always start an M-edge interchange since the middle vertex of the double edges is still covered by the other double edge. If there is an M-edge behind the crossing vertex in the direction of the path end, I can always convert the edges from there, which cannot lower the weight. The end of the path has to be omitted, because it should be short. If there are two M-edges in the interior of the circle at its beginning, it is an odd circle. You can always carry out an M-edge transformation from each of these double edges around the circle and then continue beyond the adjoining end of the path so that the weight cannot be reduced and you can break off after each M-edge at the beginning of the circle. If an M-edge is towards the beginning of the path, the other is inside, it is an even circle because it has to begin with a non-M-edge and stop with an M-edge. An M-edge conversion can be carried out from the double edge in the interior of the circle over the path end and just as with the double edges before mentioned already stop at the circle. Therefore a reduction path cannot intersect itself in a vertex. Its end cannot be an even circle. But it can return to the end vertex of a polyline in an odd circle.

3. Properties

I now use the arranged edges to compile the total graph from partial graphs by inserting a new edge. This may lead to several connected components. They single and thus their union carry a smallest matching M. The proof is by complete induction over the hitherto reached edges E_i with the vertices V_i and the corresponding matches M_i. For the sake of simplicity I leave the index i in the following proof, and write D instead of E_i. In contrast to the previously said, V with $n = |V|$, and M, always refer only to the subgraphs formed up to i.

Initial assumption.
i=2. I start with the smallest edge $e = \{v, w\}$. It possesses two vertices of degree 1, which must be matched by their edge. $\Rightarrow E_2 = M = \{e\}, V = \{v, w\}$. The matching is the smallest because it is the only one.

Induction conclusion.
i $\rightarrow$ i+1. The induction conclusion ends at the end of section 5 *"Course of the paths with smallest weight"*. I always add the next smallest edge $e = \{v, w\} \in D = E_i$ newly to D. For this I go to the list of arranged edges, skipping but the edges, which I have

at other edges already recorded in D and marked as such. Because of the continuous arrangement they are only visited once. If v or w are not from V, they are recorded newly. That is, a newly entered vertex belongs always to the smallest edge outside of D at this time. To all the vertex pairs in the extended V, I insert all the associated edges from the total graph into D. In addition to $e = \{v, w\}$ there are only $\{v, x\} \in E, \{w, x\} \in E$, with $x \in V_{old}$. These are the edges in the overall graph from the new vertices v or w to the old vertices. For this, I simply label the corresponding edges of v or w as belonging to D. In the case of an edge at v, say $\{v, y\}$, it is queried at most twice whether it should belong to D, that is to say whether is $y \in V$, namely once in the initial recording of v in V and later again when was $y \notin V$ and is added with an edge of smallest weight, such as $\{y, z\} = e$. The recording of the edges takes place in a maximum of $2|E|$ steps, a single step is needed only if two vertices v, w are added simultaneously with a next outstanding smallest edge $\{v, w\} = e$. Accordingly, the following constant properties are obtained.

(1) *Each vertex is bound to the smallest edge outside D when it is added. For $i = 1$, you should start with the empty set $D = \emptyset, V = \emptyset$.*

(2) *An edge of E with two vertices of V belongs to D. For an edge of D of course, its end vertices should belong to V. This means that every edge of a vertex that is being added is not yet in D.*

(3) *I know from the previous step a smallest matching M for D.*

(4) *A vertex x, when it is inserted, hangs at its smallest edge e_x at all, that is from E. The following useful features apply.*

(5) *The weight of the smallest matching W after increasing the new edges is not less than before that of M, $g(M) \leq g(W)$. The true small character actually occurs when the last edge used before newly inserting e, and therefore because of the ascending order all before, are strictly less than e.*

Proof. This is shown by forming a W', which is not greater than W, but for the old vertices V. I consider all the W-edges d that pass from v or w to an $x \in V, d = \{v, x\}, v \in V$, or $d = \{w, x\}, w \in V$. Then I replace d through the by (4) smallest M-edge e_x in x, $\mathrm{g}(e_x) \leq \mathrm{g}(d)$. If the smallest edge e_x occurs both for x and y, such as $\{v, x\}$ and $\{v, y\}-$ then $e_x = \{x, y\}-$ I take it only once. By this the weight only becomes smaller. The smaller edges e_x came with x into the old D and match x. This I also carry out when $x = v$ or $x = w$. Then it is the edge e with a vertex of e in V. For $v \notin V$ and $w \notin V$, I simply leave it away. This reduces the weight by $-g(e)$. This gives a matching W' of the old vertices V with old edges from D. Because M is the smallest, $\mathrm{g}(M) \leq \mathrm{g}(W') \leq \mathrm{g}(W)$ follows. If the last edge, before e is newly inserted, and therefore all e_x are smaller than e, which is less than or equal to $d = \{v, x\}$ or $d = \{w, x\}$, then $\mathrm{g}(e_x) < \mathrm{g}(e) \leq \mathrm{g}(d)$, even more if for $v \in V$ and $w \in V$ the edge e of W is omitted in W'. ■

(6) *If $v \in V$, a reduction path from a single W-edge $d = \{v, x\}, x \in V$, suffices, if the weight without g(e) can be reduced after rearrangement. The weight becomes*

strongly greater with paths from several such edges instead of e if the last edge before e used for new insertion is actually less than e. The same applies to $w \in V, f = \{w, y\}, y \in V$. *If* $v \in V$ *and* $w \in V$, *I have to try an edge both at x and y, so* $\{v, x\}$ *and* $\{w, y\}$. *One does not need an odd circle in v or w with two non-M-edges at the beginning.*

Proof. I derive (6) by selecting a further edge $h = \{v, z\}, z \in V$, besides d. h matches only v and z. For v, this also performs $d = \{v, x\}$. For z instead of h, I take the smallest e_z in the old E, this means D, $\mathrm{g}(e_z) \leq \mathrm{g}(h)$. This results in a non-larger matching W', where only $d = \{v, x\}$ is added from outside D. If the last edge used before the new insert is strictly smaller than e the same holds for all e_z and $\mathrm{g}(e_z) < \mathrm{g}(e) \leq \mathrm{g}(h)$. A further edge h adjacent to d increases the weight at most. An odd circle in v with two non-M-edges in the beginning needs there an additional M-edge for matching, so that it can already be transformed in D, this means the old edge set E, and does not reduce the weight. All this also applies to w and f. ■

In summary, (6), I have to try only one edge $d = \{v, x\}, x \in V$ instead of $e = \{v, w\}$, namely that with the largest weight loss, in addition odd circles with two M-edges at the beginning. If v and w do not belong to the old V, then I must find for both vertices edges d and f leading to the smallest weight which must be less than when e is in W.

The last property is

(7) *If only the vertex v is a newly inserted, there is no longer a weight reduction path after a short path P from* $d = \{v, x\}$ *with the largest weight loss in the previous edges. If w is also a new vertex, the same applies after another short path S of the same kind from* $y \in f = \{w, y\}$. *The smallest paths of x and y can be altered by the elimination of common partial paths in edge-foreign ones, or they can be collapsed into one.*

Proof. I limit myself temporarily to an edge $d = \{v, x\}, x \in V$. Later $f = \{w, y\}$ is similar. For the proof, we note that for a weight reducing path at the vertex x of d, only a single M-edge may be in addition to d. Otherwise, the displacement could also be performed without d in the old edge set D, and the previous matching M would not be the smallest. Because I later search the graph treelike from such single M-edges, I call the $\{v, x\}$ root edges and the vertex x at their end root vertex. In addition a downsizing path always ends at the latest in the middle vertex of a star with a non-M-edge because you can start a new path with an M-edge of the star at any time. The continuation ran completely in the old edges, and therefore cannot reduce the weight because the matching M is the smallest. However I have to stop before at a point when the weight is the least there. All new edges $\{v, z\}, z \in V$, can be replaced by edges e_z according to (6) except for the one excellent $\{v, x\}$, which I then consider as fixed. Notwithstanding I have to check each new edge $\{v, z\}$ one by one and then leave the others away. If the $\{v, z\}$ does not produce the smallest matching of all vertices, including the new one, there must of course be a path from x to z that removes the edge $\{v, z\}$ from W and inserts the edge $\{v, x\}$ in the matching W. However, since I search all $\{v, z\}$ one after the other, I also discover $\{v, x\}$

and can assume that $\{v, z\}$ is fixed and all other edges $\{v, x\}$ are disregarded or omitted. Then all paths in the old D cannot lower the weight except for those starting with only *one single M-edge* at *z, two with the odd circle*. If it is removed from M, its vertex is only matched by the fixed edge $\{v, z\}$. Without difficulty, I can start with a non-M-edge at z. It comes to M and z is matched. Such a path which is completely permitted in the old edges D can no longer produce a weight reduction, since the matching M is the smallest. It may also extend to a different root vertex, over its root edge and, if necessary, over further root vertices. Since there are no other M-edges $\{v, a\} = e$, and the path subsequently visits an edge in M and not in M, it must avoid v, because it cannot come out of v a second time, and therefore it may be performed in the old edges D, except it ends at v in an odd circle.

Suppose that Q reduces the weight after P. I state, that if P and Q collide in one vertex, they have the edge together. But they may have their initial or end vertex in common without the edge concerned. What I say in the following also applies accordingly to intersections of polylines with circles and circles with circles. If P and Q intersect in the same vertex without the following common edge, then there must be an edge of M in both before or after, a total of two, ie a star. If at least one is directed towards the path end, one can from this one always carry out a conversion of the M-edges to the path end, because of the other double edge, what cannot lower the weight, since D is at least matched. This also applies if the path Q is concerned, because after an exchange of the M-edges of P, an M-edge of P still remains at the cutting vertex. If both M-edges point towards the beginning of the path, the paths afterwards begin with non-M-edges. With such you can start any path and this cannot reduce the weight. The weight change of the whole path consists of the sum of the parts up to the cutting vertex and from there to the end. The last part, however, can only increase the weight. Because I have taken short paths, the corresponding path should have been omitted.

From this follows, they possess an edge in common, it must not belong to M. Otherwise, the first path P would change it to a non-M-edge, the second path Q would have an M-edge in front of it, so that in the intersection vertex double edges would lie at the beginning and one could change M-edges up to the path end of P, which again does not lower the weight and one has to eliminate the end.

The connnected polylines of edges which belong to P as well to Q are called R_i. Thus the paths are decomposed into partial paths P_i and Q_i, so that $P = P_1 R_1 \dots P_k R_k P_{k+1}$, $Q = Q_1 \overline{R_1} \dots Q_k \overline{R_k} Q_{k+1}$, $1 \leq k$, where the R_j represent the R_i in a different order and possibly reverse. P_1, P_{k+1}, Q_1, Q_{k+1}, R_1, and $\overline{R_1}$, and with the last two all R_i and R_j can be empty independently one from the other. The $R_i = \overline{R_j}$ are traversed twice, once by P, and once by Q and their M-edges remain unchanged. Therefore I leave the R_i, and $\overline{R_j}$ away, obtaining only edge-foreign paths P_i and Q_j.

If they have the initial or ending vertex in common, I combine them into connected components. The larger paths, because they are derived from the permissible transformation paths P and Q, are again such paths. For the sake of explanation, I look at them in detail. Because they can be traversed in opposite direction, for the end of R_i holds the same as for the beginning. Suppose the first edge of R_i belongs to M. Then the edge of

P_i before does not lie in M. After conversion over P, the said edge of P_i is contained in M and the first of R_i not. For a subsequent path Q the edge before R_i would belong to M. Thus, from this double edge, you can perform an edge transformation over R_i right at the beginning, so that the path end ought to be omitted, because I only deal with short paths. Therefore, the first edge of R_i does not belong to M and correspondingly the last and thus the edge of P_i before R_i lies in M. After applying P, the edge of P_i before R_i is removed from M and the first and correspondingly the last of R_i added in M. For the path Q, the edge of Q_j before $R_i = \overline{R_j}$ must not belong to M. After Q, R_i is back in the old state and the earlier M-edge of P_i became the M-edge of Q_j , so I can continue from P_i to Q_j. The M-edges of P_i and Q_j at R_i have just been interchanged. This also applies when the end edges of R_i lie originally in M. If the direction of the path and the edge count are reversed, P_{i+1} transits to Q_{j+1} at the end of Q_i in the correct M-edge order. This means that the paths $P_2Q_2 \ldots P_kQ_k$ continue endlessly. They just form circles, possibly several. The M-edge change along the paths P_iQ_j and $P_{i+1}Q_{j+1}$ could be performed without R_i by converting the non-M-edge at the beginning of Q_j or Q_{j+1} into an M-edge and then operating edge movements to both sides. So much to the P_i, Q_j. But this is irrelevant for further proof.

In the following cases, it can be seen that the paths can also be changed in the old edges D and the weight is not reduced because the matching M is the smallest so that the corresponding paths can then be omitted. If the single-M-edge of z, which removes P at $\{v, z\}$ from M, one of the M-edges with the odd circle, would be restored by Q, it belonged to an R_i, the remaining paths of the P_i and Q_j also ran without this edge, in the old D and could not lower the weight. You could omit the paths. Thus only P_1 uses the edge at z, such as $\{z, u\}$, and differs from Q. Also, a P_i or Q_j cannot return to the single-M-edge at z, the beginning of P_1. For single-M-edge means that all other edges at z don't belong to M, and an even circle would be formed, which can always be carried out in the old edges D. Especially, P_1 cannot form a circle with the beginning of Q_1. An even circle can always be transformed in D. If P itself represents a circle, then odd, with two M-edges at z, P_1 is to denote the piece ending at R_1, P_{k+1} the last before z. The edge $\{v, z\}$ matches z. Therefore I can begin the conversion of P_1 and P_{k+1} by removing the two M-edges at z from M. Then I proceed on both sides in P_1 and P_{k+1}. In all cases, R_1 separates P_1 and Q_1 from the remainder, R_k likewise P_{k+1} and, if present, Q_{k+1}. The paths, which consist of the remaining P_i and Q_j, can also be carried out in the old edges D and omitted for the reduction of weight, without any change in the other train of thought. This applies in particular to the above mentioned circles formed by the P_i and the Q_j. P_1, Q_1, and R_1 remain. Only R_1 and the polylines P_1Q_1 and $P_{k+1}Q_{k+1}$ remain. R_1 is not changed altogether and because of the two M-edges at z, I can convert at least one of the two polylines in D so that it cannot lower the weight and ought to be omitted. The other, such as P_1Q_1, would have to further reduce the weight by the extension Q_1, in contradiction to the fact that P was the smallest.

Thus $P = P_1R_1, Q = Q_1\overline{R_1}$. Since Q is supposed to reduce the smallest weight, it must not be empty, and thus Q_1 not or R_1 not. The path P is supposed to produce the smallest weight, means, I do not terminate before the deepest vertex, nor do I go beyond

the deepest vertex.

If R_1 is not empty, and I come with P_1 and Q_1 to opposite end vertices of R_1, $\overline{R_1}$ goes in the opposite direction of R_1 and separates P_1 and Q_1 through at least one edge. If R_1 is a circle, I can reverse the direction of R_1 and achieve $R_1 = \overline{R1}$ so that it counts to the previous case. Otherwise, P_1 is a polyline, and Q_1 is a path that is edge-foreign to P_1, which can be performed independently of P_1 and thus before P_1. Since Q_1 does not use the single-M-edge at $\{v, z\}$ because it belongs to P_1, it runs in the old edges and cannot reduce the weight. This is especially true when Q_1 returns to the initial vertex z of P_1.

If R_1 is empty and P_1 and Q_1 do not intersect in a common end vertex, they form as just now edge-free paths which can be executed in reversed order, that means Q first, so that $Q = Q_1$ does not lower the weight.

If R_1 is empty, then Q_1 is not empty and $P = P_1$ and $Q = Q_1$. If P_1 and Q_1 are combined in their end vertices, I cannot execute $Q = Q_1$ before $P = P_1$ only if Q takes the last M-edge at the common end vertex and there is no further M-edge. Then there is an M-edge of Q before the transformation, and P changes with Q in the order of alternating M-edges and is lengthened by Q. This means I have reached a lower vertex with P via the continuation Q. P was not the smallest path.

Otherwise R_1 is not empty, P and Q meet on the same end vertices of R_1, traverse it in the same direction and $R_1 = \overline{R_1}$.

Now I distinguish the following two cases.

If R_1 is not empty and Q_1 is not empty, P_1 goes to Q_1 at the initial vertex of R_1 because R_1 is deleted. Q_1 therefore lengthens P_1 as just before. I have passed P_1 through the continuation Q_1 instead of R_1 to a deeper vertex. $P = P_1R_1$ was therefore not the smallest path.

If R_1 is not empty and Q_1 is empty, then $Q = \overline{R_1}$ and $P = P_1R_1 = P_1R_1 = \overline{R_1}Q$. That is $Q = \overline{R_1}$ returns the conversion of the end of P and I have gone out with P over the deepest vertex.

It follows from all that P is not the smallest path or Q reduces not the weight. Thus, after a smallest path P, there is no decrease through a Q. The advantage of this is that for a fixed M-edge $\{v, z\}$, for an odd circle two, one has only to traverse the edges to the deepest vertex once, and then takes the smallest one for all $\{v, z\}$.

A second smallest path S from the new vertex w cannot intersect the path P. For the section is same as in the case of P and Q, that is to say it begins and ends with an edge not from M, and at S, there is no edge of M at the beginning and the end of the common path so that two edges lying one after the other not in M the path S cannot be used first. After the path conversion through P, one is allowed over S, after S, the intersection is again in the old state, it separates the end of P and S from the beginning, and one can pass at most from the beginning of P into the beginning or the end of S and from the end or beginning of S to the end of P. This means that P and S can be replaced by partial paths which originate from the starting and ending pieces starting at x and y. The remaining parts of the paths of P and S run completely in the old edges and cannot lower the weight. Thus, one can convert the smallest paths from x and y into edge-foreign

ones by omitting partial paths, or completely let them converge, so that one goes from x to y. This is the end of the proof of property (7). ■

Counterexample. *The smallest paths are not edge-foreign from the beginning.*
For this I consider the following graph consisting of two polylines $\{v,x\}P_1QP_2$ and $\{w,y\}Q_1RQ_2$ with the common polyline $R=(r)$, $\mathrm{g}(r)=7$. I represent the vertices as o with the weights of the edges in between. The fat drawn weights give the M-edges before inserting $\{v,x\}$ and $\{w,y\}$. $\mathrm{g}(e)=\mathrm{g}(\{v,w\})=18$.

P_1 : v o - 19 - $\|x$ **o - 14 - o** - 9 - **o - 10 -** **- 4 - o** -1- **o -16- o** : P_2

R : **o** − 7 − **o**

Q_1 : w o -20- $\|y$ **o -15- o** -11- **o -14- o** -8- -6- **o -5- o** -2- **o -17- o** : Q_2

The initial vertex of P_1, end vertex of P_2, initial vertex of Q_1, and end vertex of Q_2 must be matched as vertices of degree 1. The edges each bear the weight 14, 16, 15, and 17. Convince yourselves that $P=P_1RP_2$ and $Q=Q_1RQ_2$ are at least matched. The last edge of Q_2 possesses the largest weight with 17. In an ascending order, the edge e with the weight 18 is to be inserted next. With this, the edges to the already existing vertices x and y with the weights 19 and 20 are added. A reduction path from P_1 to Q_2, Q_1 to P_2, and Q_1 to Q_2 via R is forbidden because of two or three consecutive non-M-edges. Only P_1RP_2, P_1Q_1 and P_2Q_2 remain. P_1Q_1 changes the weight by $(-14+9)+(-10+8)+(-14+11)=-10$, with $\mathrm{g}(\{v,x\})=19$ in front of it becomes 9, and P_2Q_2 by $1+(-4+6)+(-5+2)=0$. The greatest weight reduction delivers $P=P_1RP_2$ by $(-14+9)+(-10+7)+(4+1)=-11$, with $\mathrm{g}(\{v,x\})=19$ it becomes 8. If one inserts only e, the weight increases by 18. After $P=P_1RP_2$, $Q=Q_1RQ_2$ becomes possible with $(-15+11)+(-14+8)+(-7+6)+(-5+2)=-14$, taking $\mathrm{g}(\{w,y\})=20$ in front of it, it results in 6. Another possibility consists in Q_1RP_2 because RP_2 gives the same weight $(-7+4)+(-1)=-4$ as RQ_2. Both paths P and Q together with the edges in front increase the weight by $8+6=14$, e alone by 18. Therefore the path transformations have to be carried out. The omission of R merges P_1 and Q_1 into a single path. The remainder P_2Q_2 has the weight change 0 as calculated above. The example shows that the lowest paths P and Q have the edge R in common and are not edge-free. They can however be replaced by edge-foreign ones, in the above example only by one, which then goes from x to y, extended from v to w. It does not matter at which new vertex v or w I start. If I start with w and y, because two or three non-M-edges follow at R, only Q_1P_1 remains with the weight decrease$(-15+11)+(-14+8)+(-10+9)=-11$ with $\mathrm{g}(\{w,y\})=20$ in front of it 9. From x, there is only the path, that removes just one edge with the weight 14 yielding the reduction -14 and with $\mathrm{g}(\{v,x\})=19$ in front of it 5. Both paths combined together with $9+5=14$ produce the same weight as above. The paths do not have a common edge and the second continues the first path to a path from y to x, extended from w to v.

4. Possible Beginnings of the Weight Reduction Paths

The case that v and w of the new edge e both belong to V cannot occur. Then their edge $\{v, w\}$ would have had to lie in D by the property (2). So $v \notin V$ or $w \notin V$.

I now need to distinguish between two different kinds of cases, one with respect to the presence of new edges at v or w one with respect to the number of new vertices v or w. With respect to the edges except *e (K0) An edge is not added at any end vertex* or *(K1) at only one end vertex* or *(K2) at both*. From $v \notin V$ or $w \notin V$ follows ($v \notin V$ and $w \notin V$) or ($v \notin V$ and $w \in V$) or ($v \in V$ and $w \notin V$). The last condition follows from the second by interchanging v and w. Thus the two cases *(F0)* $v \notin V$ *and* $w \notin V$ and *(F1)* $v \notin V$ *and* $w \in V$ remain.

(F1) $v \notin V$ *and* $w \in V$.

(K0) *An edge is not added at any end vertex.* This is a general case to which all the following cases can be traced.
If no edge is added at v, then v has the degree 1, e is a final edge, and must belong to M. For a weight reducing path, w must have a single-M-edge apart from e, with a circle two. Otherwise the reduction path could have been carried out in the old edge set D, and the previous matching would not have been the smallest. One has only to check from w all the displacement paths from this one edge, say $d = \{w, x\}$, as a unique path start, with the circle two, from the initial edge to the final edge. The details are described later.

(K1) *An edge is added to only one end vertex.* Because of $w \in V$, only at v new edges, such as $d = \{v, x, \}, x \in V$, and, if necessary, $f = \{v, y, \}, y \in V$, may arise.
1st case. I take *e to M*. Then I can omit every edge $d = \{v, x\}$ at v from M. For v is matched by $e, v \in e \in M$, and $x \in V$ has been previously matched. This edge d would only increase the weight. A displacement path over a $d \notin M$ would take d to M. Since one end vertex v is matched by e one could take the smallest edge e_x instead of d for the other end vertex x. According to the property (4), x is at its insertion at the smallest of all its edges of E, because of the selection of e_x, $\mathrm{g}(e_x) \leq \mathrm{g}(d)$. The new matching will be smaller at most. The path may also return to v in a circle via a second edge $f = \{v, y\}$. This must then be replaced by the smallest corresponding edge e_y. The improved path does not use a new edge d or f, and the previous case (K0) holds. There w can only belong to a single-edge of M besides e, in the case of a circle two, from which I must search for reduction paths.
2nd case. I leave *e away from M*. The considerations are also used in case (F0) $v \notin V$ *and* $w \notin V$. For v to be matched a $d = \{v, x, \}$, must lie in M. Because e was the smallest edge outside D, $\mathrm{g}(e) \leq \mathrm{g}(d)$, I must think of the higher weight of d. Since a d must be present, I now consider it as fixed. A displacement path that would go over $e \in M$ would take it to M. I have already dealt with this under 1st. Likewise, once again, a second edge $f = \{v, y\}$ at v, which lies in M, can be replaced by the smallest edge e_y, because of $\mathrm{g}(e_y) \leq \mathrm{g}(f)$, although the

path would eventually be decomposed into non-connnected partial paths. There remains a matching that contains d and does not use all the other edges at v such as e and f which I can conceive as not present. This results in the previous (K0) *An edge is not added at any end vertex*, but with (F1) $v \notin V$ *and* $x \in V$ relative to $d = \{v, x\}$ instead of $e = \{v, w\}$. The solution consists in taking all edges $d = \{v, x\}$ to M, where at x only one M-edge except d lies, with a circle two, and taking the decreasing displacement paths from x. There is essentially no difference between both cases if we consider $e = \{v, w\}$ as one of the edges $d = \{v, x\}$. That is, I need check all edges at v including e.

(K2) *An edge is added to both end vertices.* The case does not exist because $w \in V$ and there can be no new edge.

(F0) $v \notin V$ *and* $v \notin V$.

(K0) *An edge is not added at any end vertex.* $e = \{v, w\}$ forms its own connected component, which is at least matched by $\{e\}$, and that also with respect to the entire previous graph. I get the new matching $M + \{e\}$.

(K1) *An edge is added to only one end vertex.* This vertex is called v, its edge $d = \{v, x\}, x \in V$, and a possible further $f = \{v, y\}, y \in V$. At w there is no new edge, w has degree 1 and must be matched by $e, e \in M$. I can only change something if I take a d at v to M. Since e must lie in M and matches the vertex v of $d = \{v, x\}$, this must only be done for x. However, this is not made worse by the smallest edge e_x at the insertion of x in V because, by virtue of this property, $\mathrm{g}(e_x) \leq \mathrm{g}(d)$ holds. Therefore, each edge d can be replaced by an e_x belonging to the old edge set D, so that the path runs completely in the old edges and cannot lower the weight. This also applies if the path in a circle returns to v via an M-edge $f = \{v, y\}$. This means that this case does not occur.

(K2) *An edge is added to both end vertices.* This can be tracked back to the case (F1) of only one end vertex with (K1) new edges.
For $e \in M$, in (K1) essentially an M-edge $d = \{v, x\}, x \in V$, was replaced by the non-greater neighbor e_x of x. Here you additionally have to do it with e_y, for an M-edge $f = \{w, y\}, y \in V$. I then move entirely in the old D with the smallest matching M, where I cannot decrease the weight. The new consists in $M + \{e\}$.

For $e \notin M$, in order for v and w to be matched an $f = \{v, x\}, x \in V$, and a $f = \{w, y\}, y \in V$, must lie in M, I can then consider them as fixed and e as cancelled. For each d and f this gives the case (F1) of only one new end vertex. The edges and vertices consist in d, x, and f, y. One has to take all edges $d = \{v, x\}$ in succession to M, where at x only one M-edge except d lies, with a circle two, and start the displacement paths from x. The same applies to $f = \{w, y\}$ and y. A path may also go from such an excellent x to any y. The same applies to interchanged x and y. The smallest paths of d and f must together produce a lower weight than when I insert e alone. ■

In all cases except (F0) $v \notin V$ *and* $w \notin V$, (K0) *An edge is not added at any end vertex* and (K1) *An edge is added to only one end vertex*, the weight may possibly further be reduced. There is only a limited type of pathways that can lead to weight loss. It is always a newly added M-edge going to the old vertices V, such as $d = \{v, x\}$, except where w does not already lie in D and possesses no edge other than e. This edge d matches its end vertex x in V so that I can move an adjacent M-edge of the old D away from there. This is only possible for the first time, if there is no other M-edge there, with the circle two, otherwise I could have carried out the shift before and the matching would not have been the smallest. I consider the common vertices of such initial edges and the newly added M-edges as roots of a treelike structure to be described later in detail. I call the initial edges at x except $d = \{v, x\}$ root edges and x root vertex. For the two exceptional cases just mentioned, the M-edges except for $e = \{v, w\}$ is called the root edge and w the root vertex, e then takes the role of d. It was said above, that you have to take to M all $d = \{v, x\}$, where at x only one M-edge except d lies, with circles two, beginning with the displacement from this M-edge inclusively. Then you could do as many smallest path searches as there are edges at all, namely $|E|$. Therefore for all $\{v, x\}$ I traverse the graph simultaneously. Then one obtains as many as there are vertices, that is n. In the case of colliding paths, all are broken down except for the smallest. Finally it is shown very briefly that all $\{v, x\}$ and $\{w, y\}$ together can also be treated in one pass.

5. Course of the Paths with Smallest Weight

In order to find a smallest path, I traverse the edge set D with all the associated vertices from the new edge $e = \{v, w\}$ in a type breadth search, depending on whether an edge lies in M or not. The new edges $\{v, x\}$, and, if present, $\{w, y\}$ with the root vertices x and y, stand at the beginning and cannot be used any further. There are several x and y. In the whole of the following treatise the M-edge $e = \{v, w\}$ instead of $\{v, x\}$ must be substituted for the two exceptions, where w already lies in D, and as root vertex w instead of x. For only a single root v and later, if necessary, w, one could conceive that as a tree with additional cross edges of D. In property (7) I have proved that there are edge-foreign, smallest paths from v and w even if they are obtained by omitting cuts of two paths from v and w. However, since I do not know which edge belongs to an edge-foreign path, so that you can break the other at the meeting, I must first form a smallest path, starting say from v and then starting from w. In the previous section (4) "*Possible Beginnings of the Weight Reduction Paths*" I have fixed the $\{v, x\}$ as the beginnings of the paths. If for all these fixed $\{v, x\}$ I would preliminarily determine the smallest path, and in the end take only the smallest of all $\{v, x\}$, I could set the number of smallest path searches in the type tree with only the number of all edges $\{v, x\}$ and $\{w, y\}$, all together with $|E|$. For this reason, differently as I explained in section 4, I try to search the tree simultaneously from all edges $\{v, x\}$, take in every vertex only the lowest path and then from all $\{w, y\}$. As a result only as many tree passes arise as new vertices v, w with $e = \{v, w\}$ are created, at most n. In the end I will also treat the $\{v, x\}$ and $\{w, y\}$ at the same time.

The single-M-edges, two with the circle, which are attached to x and y and which I call root edges, I move to a non-M-edge at the first step. The root edges find themselves at the $\{v, x\}$ and $\{w, y\}$, and differ from latter. At the root edge with the son of the depth 1, therefore, hang edges not from M with the sons of depth 2, at these again all the M-edges to sons of the next step not yet in the tree, and so on. If a new edge leads to vertex, which belongs already to the tree, I take it as a non-real tree edge, as a cross edge. If their two vertices possess the same depth, I call it a horizontal edge. I insert all the vertices of the same depth with respect to the root in turn. The consequence of this is that the cross edge of a vertex z with the parent u cannot lead to a vertex arranged before u. For then z would hang on this as his father. The cross edges of z thus extend at most up to the vertices after u and before z. What is important for us alone, they have a depth by one less than z or the same, so they do not go back as far as you want.

Then I pass from such single-M-edges, with the odd circle from two, over all walkable paths, those are those where an edge $f \in M$ always is followed by an edge $h \in M$, whose affiliation to M I want to exchange, afterwards is $f \in M, h \notin M$. This traversing of the graph now depends on the matching M. With the insertion of an M-edge I can always start and end with the removal only if another M-edge matches the beginning or the end, ie at double edges or an at-least-2-star. The path ends at the latest with an edge not from M at the center of a k-star, $2 \leq k$. This consists of k M-edges with a common center. An exception is the excact-2-star at a non-M-edge, which is discussed below, because of a possible odd circle in the center. If I would convert this alone, the vertex in the origin would not be matched. At the end of a polyline with a non-M-edge at the end, it can still reduce the weight. Therefore I need to follow it in both M-edge directions. If, in other cases, I meet with an edge not from M by the path P the center of a k-star and run over one of its M-edges, I would have been able to start the following path Q immediately from the center point still matched by $k - 1$ M-edges. If the weight $\Delta\ \mathrm{g}(PQ) = \Delta\ \mathrm{g}(P) + \Delta\ \mathrm{g}(Q)$ is to be smaller than $\Delta\ \mathrm{g}(P)$, it must hold $\Delta\ \mathrm{g}(Q) < 0$.The path Q, however, runs completely in the old edges and cannot produce a smaller matching, since M is a smallest. If I hit through P with an M-edge into the center of a star, I have to take it out. If I would go further through a continuation Q by taking a non-M-edge into M, you could do this everywhere and for the same reason Q cannot lower the weight because M is already the least matched. If I terminate the path *in front of* the last M-edge, ie I leave the following star edge in M and continue the path with a different edge of the same star, the continuation path is possible in the old edges and cannot reduce the weight.

It is also excluded that I come back in an even circle to an already visited vertex, already the old edges make this circle possible. An exception is only the odd circle, when it starts and stops with two M-edges in a root vertex, as well as when it hangs at the end of a polyline. If the polyline ends with a non-M-edge and there are only the two M-edges of the circle beginning, one can convert polyline and circle because the circle start is then matched by the M-edge arising at the end of the polyline. This can lower the weight. If the circle begins with two non-M-edges, the weight of the circle alone does not decrease, since this can always be done because the circle origin is thereafter matched with at least two M-edges. But this circle can serve to match the end vertex of the polyline before the

circle when the polyline terminates with an M-edge that becomes a non-M-edge. The total weight is reduced if the weight increase of the circle is absolutely smaller than the weight reduction of the polyline. So I have to consider odd circles.

However, a reduction path can no longer go beyond the end of the circle. In both path parts through the cutting vertex, an M-edge has to lie in front or after this because of the order of the M-edges, a total of two, that is, double edges. If one is in front the cutting vertex towards the beginning of the path, the other behind the cutting vertex towards the end of the path, I can always change the path from the last because of the double edges and the path ending ought to be omitted. If the other M-edge is in the circle itself, it is an even circle, which increases the weight at most, so I should avoid it and go straight to the path end. If there is an M-edge in the circle, and the other also or outside the circle to the end of the path, I can perform an exchange of the M-edges across the circle from the M-edge in the circle and then over the path end. Therefore, when I return to a vertex that has already been visited, I break the path there. This also prevents another circle in the inner vertex of a circle. Under given circumstances one can run in the first circle only to the origin of the second circle and cease after finishing the last one. But one must consider the first independent of the second. Nevertheless at the top of the circle, several odd circles may begin and end, even with equal starting or end edges. Two M-edges at the beginning can occur only once, possibly for several odd circles, otherwise an at-least-3-star is generated. Successful paths range only from the root edges to the star edges, and correspondingly at the vertices from the root vertices to the star centers, except a 2-star is the beginning of an odd circle. Then the path stops at the first return to the origin of the circle. The paths are circle-free except for odd circles or polylines with odd circles hanging on them. Therefore I distinguish little between polylines and circles, but grasp the circles as polylines that end in the first return vertex. In this tree-like structure with the x and y as roots I look for the best reduction path. Here always edges from M and not from M are following each other. An M-edge can also lead to a depth backwards, likewise to the same depth horizontally. An edge in the same direction can also project into a star center. I walk correctly through the graph and assign the weight arisen up to here to each vertex. The weight results after an M-edge h from the previous by adding $-\mathrm{g}(h)$ because it is to be taken out, after an edge not from M by $+\mathrm{g}(h)$. I have to remember that I actually want to add the edge $\{v, x\}$ to be newly added in M and want to remove the subsequent single-M-edge. Therefore for x, I must begin with the weight $\mathrm{g}(\{v, x\})$ and not with 0 to compare all $\{v, x\}$. If, when the vertices of the graph are created, a vertex weight is entered, which exceeds all that might otherwise occur, as would be mathematically seen, one could see whether a vertex had been previously visited. For a circle, however, I must also know that the same path arrives there as before. For this purpose, and only for this, I must mark the edges with the same x_i from which the path originates, and, as I can traverse an edge forward and backward, with two x_i each. The duplicate effort is included in the visit of the vertex and is counted only once. If a path forks after an M-edge, all continuations there begin with the same weight, and the same x_i, if a path divides after a non-M-edge, only two M-edges for an odd circle may follow, since one can always perform other path conversions from

double edges in the old edges D, and therefore they cannot lower the weight. I follow the paths in both M-edge-directions. If two or more paths coincide with M-edges, that is a k-star, I can exchange the M-edges backwards to the x_i at any time, at most with $k-1$ star-M-edges. A continuation path in the star would begin with a non-M-edge, with which one can always begin. Therefore I have to stop before colliding M-edges. If edges not from M meet, I only take that one with the smallest weight and break the others. Note that if two paths run over a common M-edge $h = \{y, z\}$ and before maybe one over $y \in d \notin M$, the other over $z \in f \notin M$, then I can only use one path because, for example, with the first path is afterwards $h \notin d \in M$, it remains $f \notin M$. So I would have to go in the next path over two edges $f, h \notin M$ one after the other. As I said, I prefer the path with the smallest weight. The other path could then at most return to the old state the beginning of the first path up to d because $\in M$, $f \notin M$. In the following, d, f again denote the newly inserted edges.

According to the property (6), for the new edge $e = \{v, w\}$, a single-M-edge $d = \{v, x\}$ suffices, if necessary $f = \{w, y\}$ to generate the smallest weight. However I must try all d and f candidates. This happens first for all d, then, if necessary, the M-edges are interchanged along the smallest path, afterwards it is the turn of all f. In the following representation I restrict to $d = \{v, x\}$ and v. As stated above, I calculate the change of the path weight to the current vertex. When changing the M-edge affinity, I can run almost over every vertex. From any non-M-edge I always have to go to any M-edge and from an M-edge to a non-M-edge. The first transition is called vertex forward direction, the second vertex backward direction. This designation has the disadvantage that the direction changes continuously with two successive path vertices. I need two vertex weights in each vertex. The edge direction is less important. Since I have to go through the graph for path searching in all directions, I use notwithstanding both directions of an edge, which elevates the estimate $|D|$ to $2|D|$. I can terminate a path only by inserting an M-edge or removing a double edge.

The fact that one searches the tree from all root vertices at the same time possesses the advantage that when several paths with the same direction coincide, only the one with the least weight must be taken over and the others can be eliminated. If I enter the same direction at a vertex with a lower weight, I override the old weight and the old x_i and continue the new path in the vertex. I lock the path edge of the old path to the vertex by deleting this vertex as a successor vertex at its predecessor, the old path then ends there. To do this I have to enter its path predecessor and successor at each vertex. If I return to vertex in an odd circle, the weight can no longer be reduced and I have to abort the path. The vertex must then be marked as the start and end vertex of an odd circle. I must also enter the final weight. Then there are two additional weights in each vertex, one for each direction, four in total. I can also remember the vertex x_i from which the end of the circle comes from. But, if a circle should be formed, the x_i of the circle end must coincide with that of the beginning. Another path, which runs at a lower weight over the initial vertex finds at most minor edge weights than the old path. If he leaves the circle prematurely, a branch goes in the direction of the end vertex. A later path that traverses the whole circle runs over the same edges with at least as large weight differences as

before, and undergoes the old path at the end at least by the amount at the beginning, so that the new x_i at the end coincides with that at the beginning, if not another path over the initial vertex produces a lighter weight at all than the first at the end of the circle. A smaller weight can also come from completely different odd circles with the same two M-edges at the beginning, because only a 2-star can lower the weight, or from a circle with two non-M-edges at the beginning.

A second path which improves weight in a fixed vertex has a length at least one greater than the first. In a rough estimate a maximum of $n-1$ paths can later be added, since the length of each simple path is at most n. If the weight of the following vertices has not changed, the difference between the new path and the old path is the same, and if the previous in the next vertex was the smallest, this is transferred to the new path. If nothing else has changed, the new path after the vertex runs at least on all paths of the old one. The continuations of the previous one then disappear permanently after this vertex. This means that a path that has advanced to a vertex has never been stopped or canceled. Before I let the paths go from a fixed vertex, I determine the smallest of the paths coming from all the neighbors. The vertex weight thus becomes smaller and smaller as time passes. I now estimate how often old paths are broken off. To a previous one that was displaced, in addition at least one lower one belongs. Thus, with respect to all n vertices, at most $n/2$ paths can let die the old ones and survive themselves. I consider paths equal if they have the same length and the same vertex sequence. From these $n/2$ can survive at most the half their mutual meeting, ie $n/4$. When they come together for the ith time, no more than $n/2^i$ can survive. From $2^i \leq n$ follows $i \leq \log_2 n$ and the method ends at the latest after the $\lfloor \log_2 n \rfloor$-th generation of new vertex weights for all vertices. In the first pass the method visits at most n vertices, in fact only the successors of the root vertices x, with the second pass at most $n + n/2$. Here the double visits are counted and also when the vertex is visited for the first time at the second pass. In total, the vertex visits of the method, calculated according to the multiplicity, are at most

$$\sum_{i=0}^{\lfloor \log_2 n \rfloor} n \left(\frac{1}{2}\right)^i = n \frac{(1-(\frac{1}{2})^{\lfloor \log_2 n \rfloor + 1})}{(1-\frac{1}{2})} \leq 2n \left(1 - \frac{1}{2n}\right) = (2n-1), 1 \leq n.$$

The method ends, since the graph has only n vertices, at the latest after n passes. Then I search the smallest vertex by going through the vertex list and comparing the $4n$ values for the forward direction, the backward direction and the same at the end of the odd circle. With each vertex visit with a smaller weight, I have the last predecessor vertex noted and the other predecessor vertices, that is, the access edges of that one, blocked. I run the path in the success case in at most n steps from the smallest vertex back to the root vertex x and change with it the M-edges and non-M-edges. If a short path is required, I have to take the last smallest vertex from the end and get the first from the beginning. The return to the root also confirms from which x the path comes from. For this I need no more than n steps, with the determination of the smallest weight $4n + n = 5n$. On the first visit of the at most n vertices I need because I have to work with directed edges, $2|D|$ edges. This includes the fact that in the forward direction and reverse direction different M-edges and non-M-edges are visited in the same vertex. For the repeated vertex visits

one can also not estimate this better, in total therefore with $2(\log_2 n)|D|$. Finding the smallest path followed by changing the M-edges to the smallest matching does not take more than $2(\log_2 n)|D| + 5n$ steps.

Note the vertex weight in addition to polylines for the odd circles as well. Otherwise they are treated as simple paths. If a path from the root x does not lower the weight, I have to leave everything as it is. The weight change including $e = \{v, w\}$ is then $+g(\{v, w\})$, corresponding to y. If an edge from the new vertices v and w leads to the same x, that is, $\{v, w, x\}$ forms a triangle, then the path counts to x, and if the path from x does not lead to the smallest weight, a path may later start once again at $y = x$.

The effort involved is that I first search the kind of tree from the many roots x where I can break at at-least-3-stars, and then, if necessary, exchange the M-edges. All together we get $2(\log_2 n)|D|+5n$. Thus I obtain the smallest matching for the next edge set E_{i+1} consisting of the old edges $D = E_i$ and the newly added edges $\{v, x\}$, where v comes to V_{i+1}. Later E_{i+2} results from E_{i+1} with the additional edges $\{w, y\}$. With w and E_{i+1}, I can still change the edges $\{v, x\}$, thus finally form a path from w to v. In the end, $\{v, w\}$ is inserted into E_{i+2} and, if the weight is smaller, only $\{v, w\}$ is taken up in M instead of all M-edge changes of E_i and E_{i+1}. In this way each vertex v_i is newly added exactly once. The new set E_i then contains v_i.

Here the induction conclusion ends, which started at the beginning of section 3 "*Properties*". ■

Now consider the set of all edges E and call the set of all vertices now V again with $n = |V|$. In retrospect I begin with an edge e=$\{v, w\}$ with two vertices. In the remaining steps, a new edge comes to the old edge set E_i, therewith a new vertex v_i, then possibly a vertex $w_i = v_{i+1}$. The effort in the tree does not depend very much on the number of $\{v_i, x\}$ or $\{w_i, y\}$ because the heavier one is broken when the paths collide. I have inserted v_i and w_i one after the other. This can be performed at the same time for fixed i. The reasoning is as follows. The v-paths and w-paths are distinguished by the identifiers x and y, so they can be separated. If they collide in two M-edges or non-M-edges, the heavier is broken off. But that does not matter if they are executed consecutively. When w is inserted after v, the v-path and the w-path are successively converted. This also holds in the case of the reasoning of property (7) in section 3 "*Properties*", for intersecting paths P and Q: If the cutting edge does not lie in M, the edge before it in the first path must belong to M, the one before it in the second path not. If the cutting edge lies in M, it is reversed. In either case, I can continue the initial part of the first path in the second path to a single path. The cutting edge is converted twice, so it separates the beginnings from the ends. The ends are full in the old edges D and cannot lower the weight. The two reduction paths can be replaced by one. The second path cannot be executed simultaneously, that is to say independently of the other, because it would have to pass over two non-M-edges or two M-edges. Therefore he discovers the only one path to the other starting vertex x with opposite direction in the second path part. If executing one after the other the original paths are the smallest, now the united path is the smallest. This case of a single path should always occur when the smallest downsizing paths intersect. Otherwise, I get two edge-foreign. A good explanation of this is provided

in the counterexample at the end of section 3 "*Properties*". There, the first path P_1RP_2 of x thereafter allows two exclusive paths Q_1RQ_2 and Q_1RP_2 of y. The paths of x and y intersect in R. The result forms the path P_1Q_1 from x to y.

I now count all the steps together. For an edge $\{v_i, v_{i+1}\}$ without further edges at v_i and v_{i+1}, I need no path search. I start with $v_1, E_1 = \emptyset$ and $v_2, E_2 = \{\{v_1, v_2\}\} = \{e\}$. Then according to the method I form a smallest matching for E_3 with E_2 as the old edge set. In general, for E_{i+1} there is no more than $2(\log_2 i)|E_i| + 5$ steps because of $|V_i| = i, 2 \leq i, i \leq n-1$. So I have to repeat the procedure at most $(n-2)$ times. I estimate $|E_i| \leq |E|$ and obtain a total

$$\sum_{i=2}^{n-1}(2(\log_2 i)|E_i| + 5i) \leq \sum_{i=2}^{n-1}(2(\log_2 i)|E| + 5i) \leq 2(\log_2((n-1)!))|E| +$$

$$5\left(\frac{n(n-1)}{2} - 1\right) < 2(\log_2 n)(n-1)|E| + 5\frac{n(n-1)}{2}, \quad 3 \leq n.$$

The last expression gives the value $2+5 = 7$ for $n = 2$ because $|E_1| = 1$. The true value is 0 because I do not have to do anything with the first edge. The estimate is therefore true for $2 \leq n$.

Finally I give the effort for the whole procedure down to the constants with which you have to multiply. I need at most $2|E|$ for the conversion to a suitable input in the beginning of section 1,
$(\log_2 |E| + 1)|E| \leq (\log_2 n(n-1)/2) + 1)|E| = (\log_2 n(n-1))|E| \leq$
$2(n-1)|E|, 2 \leq n$, for the ascending arrangement of the edges in the end of section 1 because of $\log_2 n(n-1) \leq 2(n-1)$ due to $n \leq 2^{n-1}, 0 \leq n$, and $n(n-1) < n^2 \leq 2^{2(n-1)}, 2 \leq n$,
$2(\log_2 n)(n-1)|E| + 5n(n-1)/2$, for the last treated least matchings of the edge by edge increased subgraphs.

From the sum it follows because of $n \leq 2|E|$ the assertion of the essay

$$2|E| + 2(n-1)|E| + 2(\log_2 n)(n-1)|E| + 5\frac{n(n-1)}{2} \leq$$

$$(2 + 2 + 2(\log_2 n) + 5)(n-1)|E| =$$

$$(9 + 2\log_2 n)(n-1)|E| \leq 11(\log_2 n)(n-1)|E|, 2 \leq n.$$