

THE UNIVERSITY OF CHICAGO

Figure 1 illustrates the steps of the proposed algorithm for finding a minimum spanning tree. The process starts with a graph with 10 nodes and 15 edges. The algorithm iteratively selects edges with the minimum weight, avoiding cycles and ensuring no node has a degree greater than 2. The steps are as follows:

- (a) Initial graph with 10 nodes and 15 edges.
- (b) Select edge (1,2) with weight 1.
- (c) Select edge (2,3) with weight 1.
- (d) Select edge (3,4) with weight 1.
- (e) Select edge (4,5) with weight 1.
- (f) Select edge (5,6) with weight 1.
- (g) Select edge (6,7) with weight 1.
- (h) Select edge (7,8) with weight 1.
- (i) Select edge (8,9) with weight 1.
- (j) Select edge (9,10) with weight 1.
- (k) Select edge (1,3) with weight 2.
- (l) Final minimum spanning tree with 9 edges and total weight 9.

Figure 1 consists of several Feynman diagrams illustrating the production of a Z boson and a photon in the decay of a top quark. The diagrams are arranged in two columns. The left column shows tree-level diagrams, and the right column shows loop-level diagrams. The diagrams involve top quarks, bottom quarks, and various gauge bosons (W, Z, photon). The diagrams are labeled with 'a' through 'j'.

Year	United States (%)	Japan (%)	Germany (%)
1950	15	15	25
1960	16	18	28
1970	17	22	30
1980	18	28	32
1990	19	32	33
2000	20	34	34
2010	21	35	35
2020	22	35	35
2030	23	35	35
2040	24	35	35
2050	25	35	35

Figure 1: Schematic representation of the proposed model. The diagram shows a vertical stack of layers. At the top, a layer of 'Photons' is shown with wavy arrows. Below this is a layer of 'Photons' with wavy arrows and 'Electrons' with small circles. This is followed by a layer of 'Photons' with wavy arrows and 'Electrons' with small circles. Below this is a layer of 'Photons' with wavy arrows and 'Electrons' with small circles. At the bottom, a layer of 'Photons' with wavy arrows is shown. The layers are labeled on the right as 'Photons', 'Photons', 'Photons', 'Photons', and 'Photons'.

SRI.gov.ec