I use ford fulkerson algorithm as base algorithm and I use BFS to find an augmenting path.

Time complextiy:
The time complexity for my solution is O(VE^2), because every time we run BFS to search a augmenting path, it will reduce revert a edge on residual graph, suppose that edge is (v1 -> v2) and reverted to (v2 -> v1). And originally the distance from source to v1 and v2 satisfifys d(source, v1) + 1 = d(source, v2). If during the loop we will need to revert the edge again (the augmenting path has (v2->v1) in it), then we will have d'(source, v2) + 1= d'(source, v1). And because d'(source, v1) = d'(source, v2) + 1 >= d(source, v2) + 1= d(source, v1) + 2, so once we revert the edge again, the distance will at least increase by two. But the augmenting path's length is limited to O(E), so the increase will not increase O(E/2) times. And we have O(V) vertices, Therefore the fulkerson algortihm with BFS will at most run O(VE) times. And for each time we search for a augmenting path, we will need O(V + E) = O(E) times to perform BFS, The overall time would be O(VE^2)

Correctness Proof:
The Ford Fulkerson Algorithm is defined as A, and we assume there's a most similar algorithm A' but got a best result. And the better result has at least one pair of day, event pair differs from the A's result. For that event_j, result A has day_i -> event_j, an result A' has day_k -> event_j.

Now we will construct an algorithm A'' by change the result A' of result from day_k -> event_j to day_i -> event_j. And there will be two cases:
If day_i is not used in the result of A', then then after the change, the maximum events can be attended mains same for A'' as the only difference is which day event_j will be visited. So A'' can get the best result but it is more similiar to A than A', which conflicts our assumption that A' is the most similar algorithm.
If day_i is noted in the result of A', say day_i -> event_a. Now we repeat the above process, if on result A there's day_g->event_e, we change the day_g on result A' to day_g -> event_e. And if there left a event_b again, we repeat the process. The process will be stop for two cases:

1. we end up have a day is unused in A', which is same as we proofed above, the A'' will be simliar to A than A', our correctness holds.
2. we end up have a day_x that only can attend event_y in A', and if we change day_x to connect to event_z according to A, event_y will not be attended. But this will never happen because if we have event_x -> event_y and we did not revert the edge between event_x->event_y, there will be a augmenting path from source->event_x->event_y->sink, which contradicts that our algorithm will end if we cannot find a augmenting path. So our correctness holds

Therefore, Ford Fulkerson Algorithm is correct for this problem