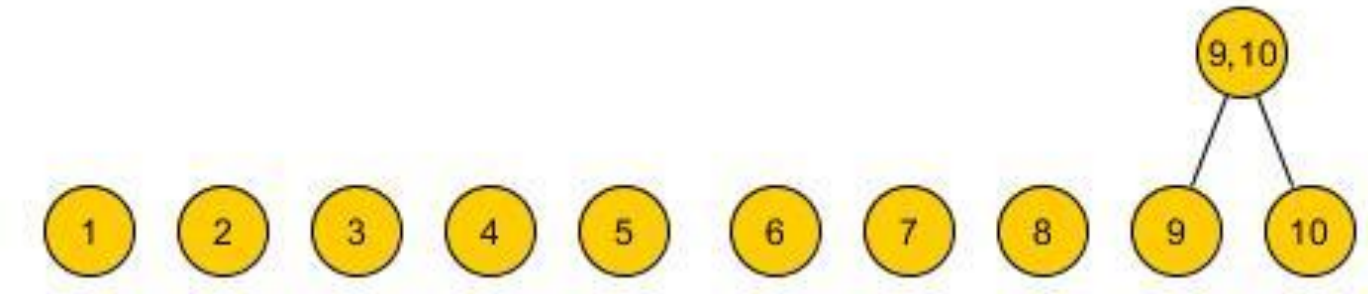
Building the Linkage Tree (LT) in LTGA

1. Start with singleton linkage sets

Thierens, D. (2010). The linkage tree genetic algorithm. In *Parallel Problem Solving from Nature, PPSN XI* (pp. 264-273). Springer Berlin Heidelberg.

Building the Linkage Tree (LT) in LTGA

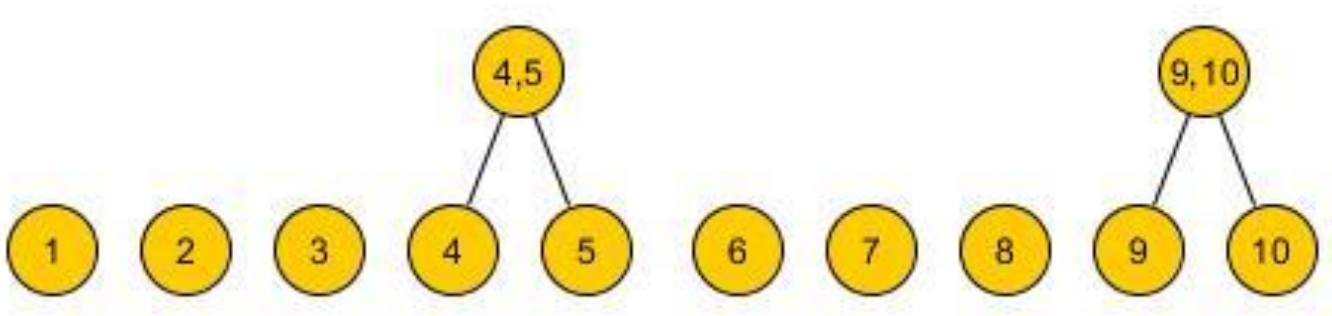
2. Compute MI for all pairs of clusters
3. Cluster 2 sets with the highest MI
4. Repeat steps 2 and 3 until 2 cluster remain



Thierens, D. (2010). The linkage tree genetic algorithm. In *Parallel Problem Solving from Nature, PPSN XI* (pp. 264-273). Springer Berlin Heidelberg.

Building the Linkage Tree (LT) in LTGA

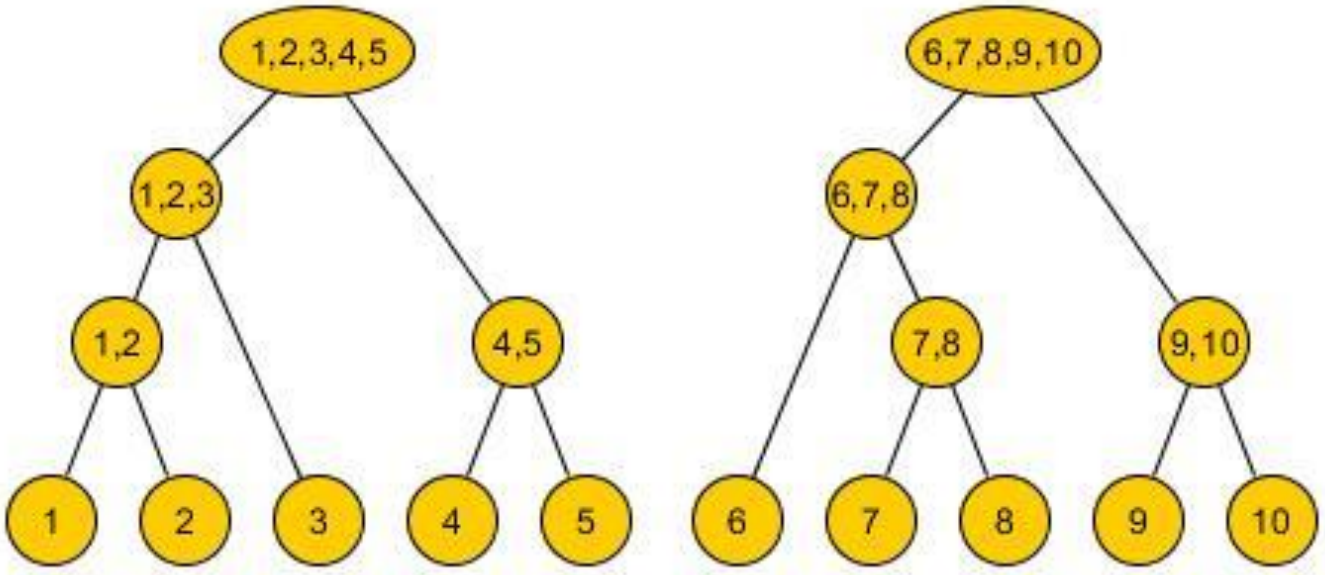
2. Compute MI for all pairs of clusters
3. Cluster 2 sets with the highest MI
4. Repeat steps 2 and 3 until 2 cluster remain



Thierens, D. (2010). The linkage tree genetic algorithm. In *Parallel Problem Solving from Nature, PPSN XI* (pp. 264-273). Springer Berlin Heidelberg.

Building the Linkage Tree (LT) in LTGA

2. Compute MI for all pairs of clusters
3. Cluster 2 sets with the highest MI
4. Repeat steps 2 and 3 until 2 cluster remain

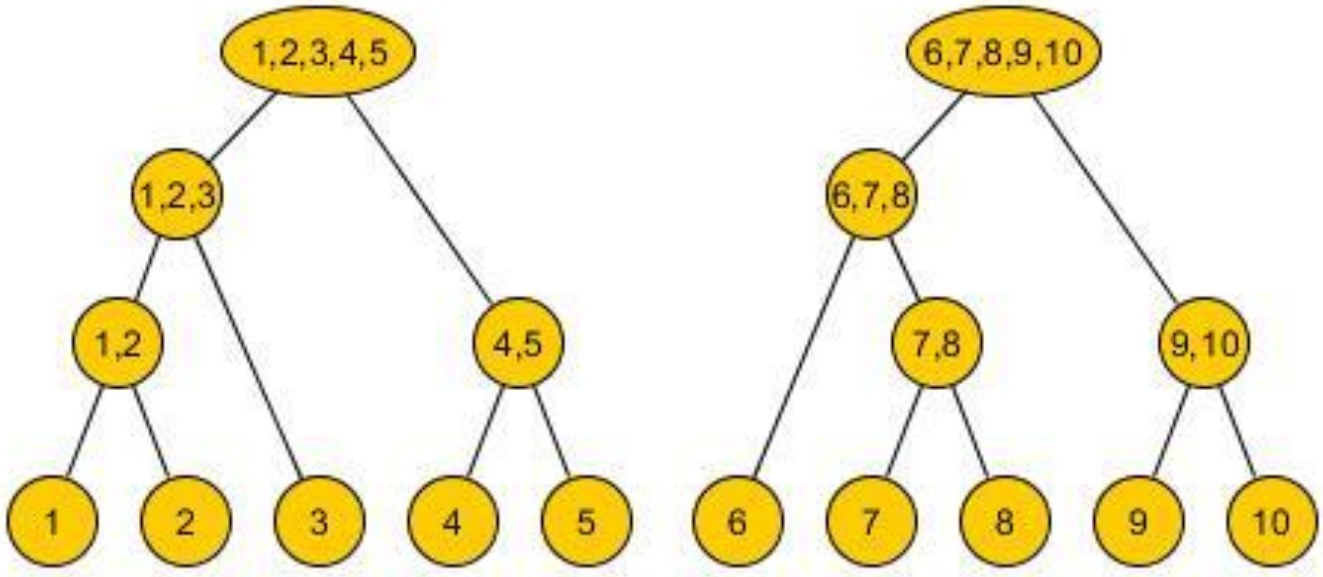


Thierens, D. (2010). The linkage tree genetic algorithm. In *Parallel Problem Solving from Nature, PPSN XI* (pp. 264-273). Springer Berlin Heidelberg.

Optimal Mixing (OM) in LTGA

- For each individual p_i in the population
- Traverse all masks in the Linkage Tree

p_i	3	3	2	2	1	-1	0	0	0	0



Thierens, D., & Bosman, P. A. (2011, July). Optimal mixing evolutionary algorithms. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation* (pp. 617-624). ACM.

- For each mask (in reversed order of merging), randomly select a parent p from the population
- Donate values of the variables in the mask from the parent to p_i
- If this leads to an improvement, continue the search with the updated solution

p_i	3	3	2	2	1	-1	0	0	0	0

Initial solution (5 mil)

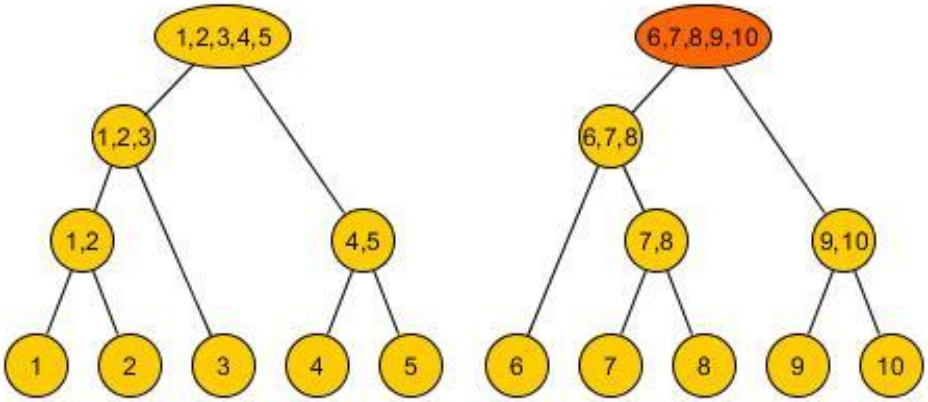
+

p	3	0	1	2	1	-1	0	0	2	0

Random parent

o_i	3	3	2	2	1	-1	0	0	2	0

Improved offspring (4.8 mil)



Thierens, D., & Bosman, P. A. (2011, July). Optimal mixing evolutionary algorithms. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation* (pp. 617-624). ACM.

- For each mask (in reversed order of merging), randomly select a parent *p* from the population
- Donate values of the variables in the mask from the parent to p_i
- If this leads to an improvement, continue the search with the updated solution

o_i	3	3	2	2	1	-1	0	0	2	0

Intermediate solution (4.8 mil)

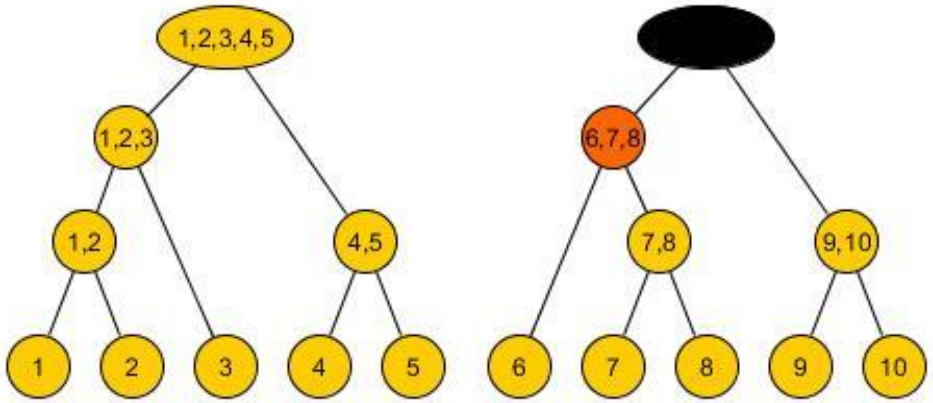
+

p	0	0	0	1	1	2	-2	-2	1	1

Random parent

	3	3	2	2	1	2	-2	2	2	0

Infeasible solution



Thierens, D., & Bosman, P. A. (2011, July). Optimal mixing evolutionary algorithms. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation* (pp. 617-624). ACM.

- For each mask (in reversed order of merging), randomly select a parent *p* from the population
- Donate values of the variables in the mask from the parent to p_i
- If this leads to an improvement, continue the search with the updated solution

o_i	3	3	2	2	1	-1	0	0	2	0

Intermediate solution (4.8 mil)

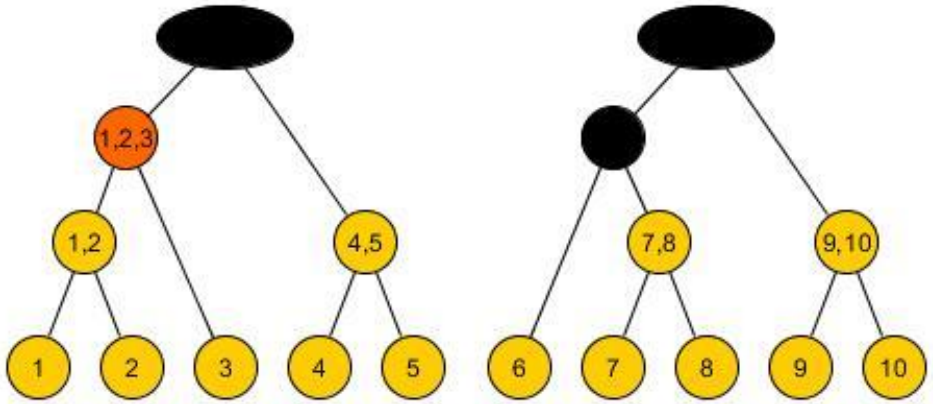
+

p	3	3	3	1	1	0	0	0	0	0

Random parent

	3	3	3	2	1	-1	0	0	2	0

Worse offspring (6 mil)



Thierens, D., & Bosman, P. A. (2011, July). Optimal mixing evolutionary algorithms. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation* (pp. 617-624). ACM.

- For each mask (in reversed order of merging), randomly select a parent *p* from the population
- Donate values of the variables in the mask from the parent to p_i
- If this leads to an improvement, continue the search with the updated solution

o_i	3	3	2	2	1	-1	0	0	2	0

Intermediate solution (4.8 mil)

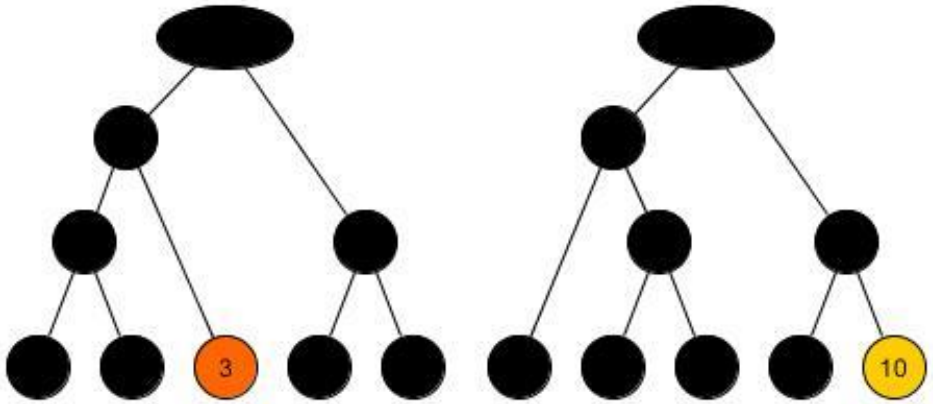
+

p	3	3	1	1	-1	1	2	0	1	0

Random parent

o_i	3	3	1	2	1	-1	0	0	2	0

Improved offspring (4 mil)



Thierens, D., & Bosman, P. A. (2011, July). Optimal mixing evolutionary algorithms. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation* (pp. 617-624). ACM.

Linkage Tree Genetic Algorithm (LTGA)

```
    for all i ∈ {0, 1, ..., n − 1} do
      P_i ← RandomSolution(n)
    end for
    while Termination criteria not met do
 5:   if F not predetermined then
        F ← BuildLT(P)
      end if
      for all i ∈ {0, 1, ..., n − 1} do
        b ← o ← P_i
10:     fitness[b] ← fitness[o] ← fitness[P_i]
        for all j ∈ {0, 1, ..., |F| − 1} do
          p ← Random({P_0, ..., P_{n−1}})
          o_{F^i} ← p_{F^i}
          if o_{F^i} ≠ b_{f^i} then
15:         EvaluateFitness(o)
            if fitness[o] > fitness[b] then
              b_{f^i} ← o_{f^i}
              fitness[b] ← fitness[o]
            else
20:           o_{f^i} ← b_{f^i}
              fitness[o] ← fitness[b]
            end if
          end if
        end for
25:     O_i ← o
      end for
      P ← TournamentSelection(O, n, 2)
    end while
```

Thierens, D. (2010). The linkage tree genetic algorithm. In *Parallel Problem Solving from Nature, PPSN XI* (pp. 264-273). Springer Berlin Heidelberg.