Knapsack Problem

of John, Inga, Lune, Nora, Tillmann

Our Implementation

Defining a Knapsack-Case

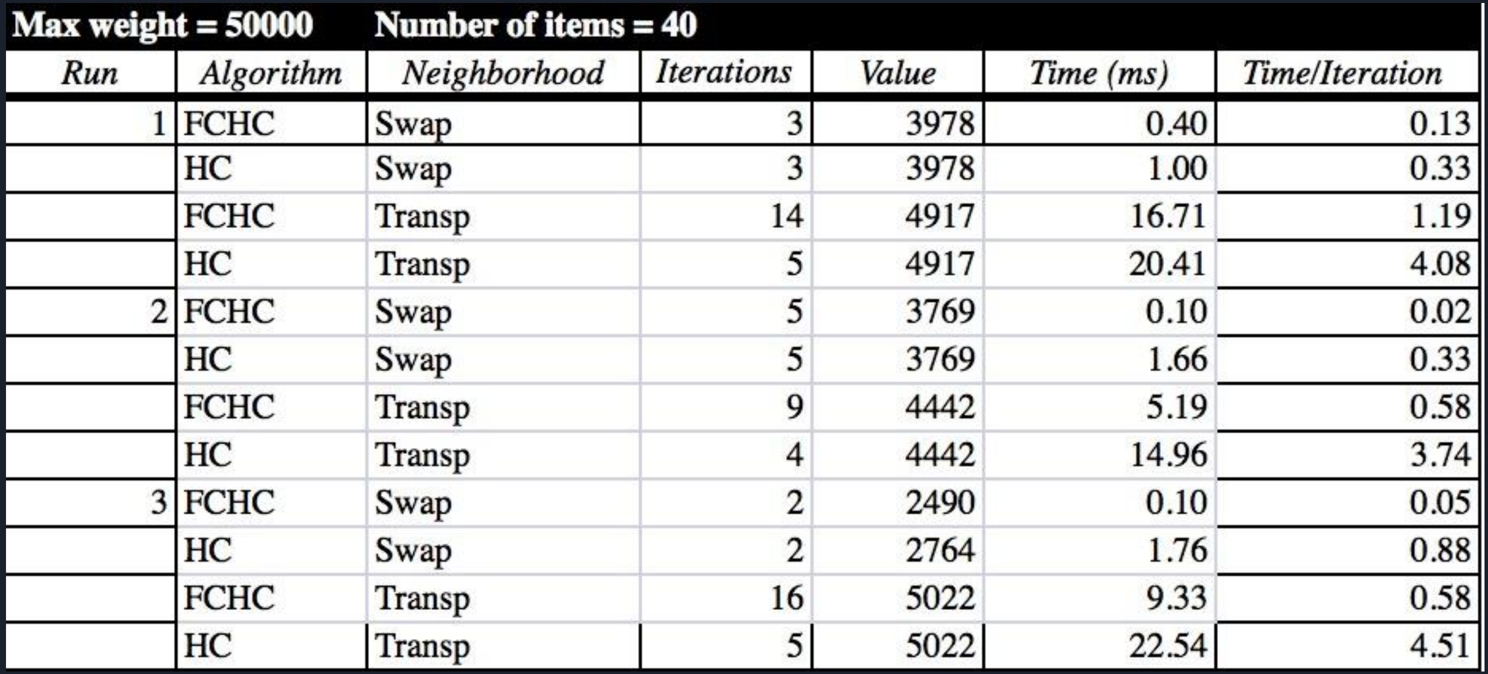
- fixed number of items
- fixed maximum weight of the knapsack
- randomly initialized values and weights
- item can be in the bag or not (boolean)

We defined two kinds of Neighborhoods

- smaller swap neighborhood
- larger transpositional neighborhood

Our Results

runs with 40 items and maximum weight of 50,000



Max weight = 50000		Number of items = 40				
Run	*Algorithm*	*Neighborhood*	*Iterations*	*Value*	*Time (ms)*	*Time/Iteration*
1	FCHC	Swap	3	3978	0.40	0.13
	HC	Swap	3	3978	1.00	0.33
	FCHC	Transp	14	4917	16.71	1.19
	HC	Transp	5	4917	20.41	4.08
2	FCHC	Swap	5	3769	0.10	0.02
	HC	Swap	5	3769	1.66	0.33
	FCHC	Transp	9	4442	5.19	0.58
	HC	Transp	4	4442	14.96	3.74
3	FCHC	Swap	2	2490	0.10	0.05
	HC	Swap	2	2764	1.76	0.88
	FCHC	Transp	16	5022	9.33	0.58
	HC	Transp	5	5022	22.54	4.51

Resume

Swap vs Transpositional Neighborhood:

- The algorithms are faster for swap neighborhood
- Need less steps for swap neighborhood
- Transpositional neighborhood finds better solutions

First Choice Hillclimb vs Hillclimb:

- FCHC needs more steps than HC
- But FCHC is faster than regular HC
- FCHC almost always gives equally good solutions

Thank you!

AFTER CLIMBING A GREAT HILL
ONE ONLY FINDS
THAT THERE ARE MANY MORE
HILLS TO CLIMB
-NELSON MANDELA