Updated distribution algorithm: Determine how many jobs a job owner of interest should try to acquire, provided there is a set of such owners sharing a common pool of jobs, in which some arbitrary number of jobs is available for acquisition.

Basic idea: maintain a subset of job owners in which all owners are assigned the same number of jobs.
Grow the subset by adding owners with larger number of jobs by equating number of jobs of owners from minimal subset to the new higher number by assigning new jobs from available jobs pool to each owner of minimal subset.
If there is not enough jobs to make numbers equal, distribute as much as possible equally between the owners from minimal subset. If owner of interest is present in minimal subset, and non-zero remainder of jobs exists, give 1 more job to the owner of interest.

Job limit: If some job owners have limit of jobs, they need to be removed from minimal subset upon reaching the limit.
In this case each iteration should choose the closest job number count as its goal, among job limits of job in the minimal subset and next minimal job number for owners not present in the minimal subset

Old jobs: Jobs that haven't been loaded after a certain timeout (4×pulse) should be considered unconditional targets.
That means they should be excluded from distribution algorithm and any server should attempt to load them all.

Available jobs
Minimal subset

5	10	12	25	17
5+5	10			12
5+5+2	10+2	12		8
5+5+2+2	10+2+2	12+2		2
9+1	4+1	2+1	0	

10	12	15	25	6
10+2	12			4
10+2+2	12+2			0
4+0	2+0	0	0	

10	14	15	25	3
10+3				0
3+0				

Available jobs

Limit reached

Minimal subset

5 (50)	10 (13)	12 (16)	25 (50)	17
5 +5	10			12
5 +5+2	10 +2	12		8
5+5+2+1	10+2+1	12+1		5
5+5+2 +1+2		12 +1 + 2		1
10+1	3	3+1	0	

5 (50)	10 (13)	12 (13)	25 (50)	17
5 +5	10			12
5 +5+2	10 +2	12		8
5+5+2+1	10+2+1	12+1		5
5+5+2 +1 +5				1
13	3	1	0	