Algorithm:

1. Create a network N with vertices $V = \{s, a_1, \dots, a_n, b_1, \dots, b_n, t\}$ and edges
   - $E = \{(s, a_1), \dots, (s, a_n)\}$ (with $c(s, a_i) = b$)
   - $\cup \{(b_1, t), \dots, (b_n, t)\}$ (with $c(b_j, t) = r$)
   - $\cup \{(a_i, b_j) : d(s_i, s_j) \leqslant d\}$ (with $c(a_i, b_j) = 1$).
2. Find a maximum integer flow f in network N (using the Edmonds-Karp algorithm, for example).
3. If $|f| < rn$, then return NIL; else, for $j = 1, \dots, n$, set $B_j = \{s_i : f(a_i, b_j) = 1\}$, and return $B_1, \dots, B_n$.

Correctness: Every collection of backup sets $B_1, \dots, B_n$ with no sensor belonging to more than b sets and no backup set of size more than r (but not all B_j necessarily having size r) gives rise to a flow f in N as follows:

- $f(s, a_i)$ = number of backup sets that sensor s_i belongs to (not more than b so edge capacity respected);
- $f(a_i, b_j) = 1$ iff $s_i \in B_j$;
- $f(b_j, t)$ = number of sensors in backup set B_j (not more than r so edge capacity respected).

In this way, flow is conserved at each a_i because the total flow out is exactly equal to the number of backup sets that s_i belongs to, and flow is conserved at each b_j because the total flow in is exactly equal to the number of sensors in backup set B_j. So the maximum flow value $|f|$ is at least as large as the total size of all the backup sets.

Every integer flow in N gives rise to a collection of backup sets $B_1, \dots, B_n$ with no sensor belonging to more than b sets, as follows:

- $B_j = \{s_i : f(a_i, b_j) = 1\}$;
- no sensor belongs to more than b backup sets because $c(s, a_i) = b$;

For these backup sets, $|B_1| + \cdots + |B_n| = |f|$ because both sides are equal to the number of edges (a_i, b_j) with $f(a_i, b_j) = 1$. This means that the total size of all the backup sets is at least as large as the maximum flow value in N.

Hence, finding a maximum flow in N yields a collection of backup sets with the maximum total size. If this size is equal to rn, then the backup sets $B_1, \dots, B_n$ defined above satisfy the conditions of the problem; else, there is no collection of backup sets that is large enough.

Runtime: Creating N takes time $\Theta(n^2)$ (every pair of sensors (s_i, s_j) must be examined); solving the maximum flow problem takes time $\Theta(n^5)$ (using the Edmonds-Karp algorithm); constructing the backup sets takes time $\Theta(n^2)$ (every edge (a_i, b_j) must be examined). The total is $\Theta(n^5)$—dominated by the time to solve the maximum flow problem.