Задача состоит в поиске максимального потока минимальной стоимости. Идея алгоритма состоит в построении на основе эвакуационного плана сети, где здания и бункеры будут вершинами. Построим из предложенного плана эвакуации остаточную сеть. Если в ней есть отрицательный цикл (критерий оптимальности) - предложенный план эвакуации не оптимален, если нет - увеличим поток вдоль каждого ребра цикла на 1 и повторим процесс проверки. Циклы будем искать алгоритмом Беллмана-Форда.

Сложность алгоритма
Первым этапом мы инициализируем сеть. Сначала определяем стоимость прохода по ребрам, подсчитывая расстояние от каждого здания до каждого бункера. Сложность - O(n*m), n - число зданий, m - число бункеров (подсчет расстояния между двумя точками занимает O(1)). Затем считываем предложенный эвакуационный план, попутно строя остаточную сеть. Сложность все та же - O(n*m).
Вторым этапом мы циклично проверяем сеть на наличие отрицательных циклов и удаляем их.
Проверка на наличие отрицательных циклов происходит с помощью алгоритма Беллмана-Форда. Для каждой вершины мы проверяем все исходящие из нее ребра и смотрим, нашли ли мы путь меньшей длины. Сравнение пути и добавление очередной вершин в очередь занимает константное время, а значит сложность ограничивается O(V*E). В нашем случае, V = (n+m+2), E = (n+m+2)*(n+m+2) (мы храним ребра в двумерном массиве), т.е. O((n+m)^3).
Удаление отрицательного цикла ведется последовательным увеличением потока вдоль каждого ребра отрицательного цикла. Стоимость нового потока таким образом уменьшится как минимум на 1. Стоимость всего цикла = длина цикла*стоимость самого дорого ребра*пропускная способность ребра = O((n+m)*COST*MAXCAP).
Итого, сложность алгоритма составляет O(n*m) + O((n+m)^3)*O((n+m)*COST*MAXCAP) = O(((n+m)^4)*COST*MAXCAP).

Корректность алгоритма
Исходит из корректности критерия оптимальности и корректности алгоритма Беллмана-Форда поиска цикла отрицательного веса. Процесс удаления отрицательного цикла корректен, т.к. в ходе одной итерации его стоимость уменьшается хотя бы на единицу и конечен, т.к. стоимость цикла ограничена величиной длина цикла*стоимость самого дорого ребра*пропускная способность ребра.

Корректность критерия оптимальности:
Поток оптимален <=> остаточная сеть не содержит отрицательных циклов
Необходимость: Пусть поток оптимален и в остаточной сети есть цикл отрицательного веса. Пустим по нему добавочный поток, равный минимальной остаточной пропускной способности ребер цикла. Величина потока осталась прежней, а стоимость уменьшилась => поток не оптимален.
Достаточность: Используем вспомогательные леммы
Лемма 1: поток представим в виде совокупности путей из истока в сток и циклов, имеющих положительный поток.
Док-во:
Пусть величина потока > 0. Тогда из истока выходит хотя бы одно положительное ребро. Если оно ведет в сток, то мы нашли нужный путь. Если нет, то из него согласно свойству сохранения потока исходит хотя бы одно ребро, имеющее положительный поток. Повторяя процесс прохода, мы либо попадем в сток (нашли путь), либо вернемся в прежнюю (нашли цикл). Уменьшим поток вдоль найденного пути/цикла на величину минимального потока его ребер. Повторив процесс по вновь полученному потоку, мы рано или поздно уменьшим поток вдоль всех ребер до 0, тем самым найдя искомую декомпозицию потока на пути и циклы.
Лемма 2: Пусть имеются два потока f и g одной величины. Тогда g представим в виде f + несколько (возможно, 0) циклов в G_f.
Док-во:
Вычтем f из g и получим некоторый нулевой поток. Проведем его декомпозицию по Лемме 1. Т.к. поток нулевой, то путей в нем нет. Значит, в нем есть только циклы. Но циклы в сети G являются циклами и в сети G_f. Значит мы нашли искомое представление.

Рассмотрим теперь поток f, в остаточной сети которого нет отрицательных циклов и поток оптимальный f* той же величины. По лемме 2 f* представим в виде f + несколько циклов. Стоимости этих циклов >= 0, а значит стоимость f* >= стоимости f. С другой стороны, f* оптимален, а значит <=. Таким образом, стоимость f* = стоимости f, и f также оптимален.

Корректность алгоритма Беллмана-Форда для поиска циклов отрицательного веса.
Пусть граф G содержит содержит цикл отрицательного веса, достижимый из стартовой вершины, т.е. сумма весов вдоль цикла отрицательна. Пусть на последней итерации алгоритма стоимость прохода в каждую вершину не уменьшилась, т.е.
$d[v_i] \leq d[v_{i-1}] + \omega(v_{i-1}, v_i)$ для любой вершины цикла.
Просуммируем по циклу:

$$\sum_{i=1}^{k} d[v_i] \leq \sum_{i=1}^{k} d[v_{i-1}] + \sum_{i=1}^{k} \omega(v_{i-1}, v_i)$$

Т.к. $v_0 = v_k$, то $\sum_{i=1}^{k} d[v_i] = \sum_{i=1}^{k} d[v_{i-1}]$. Следовательно, $0 \leq \sum_{i=1}^{k} \omega(v_{i-1}, v_i)$, т.е. цикл неотрицателен. Противоречие.