Общие замечания. Для начала заметим что любая команда из 2-ух и менее человек подходит под ограничения задачи и всегда можно выбрать 1 или 2 игроков с максимальными эффективностями. Пусть теперь наша оптимальная команда состоит из $n > 2$ игроков. Среди них есть два с минимальной эффективностью, пусть их суммарная эфферктивность E_min, а их эффективности E_1 и E_2 соответственно, тогда в эту комманду можно включить любого игрока с эффективностью лежащей в отрезке $[E_2; E_min]$.

Утверждение. Пусть $E_1 ... E_n$ отсортированный по неубыванию список эффективностей игроков, тогда оптимальный ответ на задачу достигается на каком-либо подотрезке этого списка.
Доказательство. Допустим что оптимальный ответ состоит из $n > 2$ игроков (случай меньшего количества разобран выше). И стреди их эффективностей в отсортириванном списке имеются проруски. Пусть есть пропущенный игрок с индексом j в отсортированном списке, тогда все игроки включенные в команду имеют эффективность меньше либо равную E_j. Выберем вместо этих игроков игроков с индексами на 1 больше тогда суммарная эффективность стала не меньше и сумма двух минимальных не уменьшилась, т.е. условие сплоченности не нарушилось, и суммарная эфферктивность такой команды не хуже оптимальной. Таким образом му можем произвести такие сдвиги игроков и получить команду, которая является подотрезком в отсортированном списке эффективностей и суммарная их эффективность не уменьшится.

В итоге задача свелась к следующей - нахождение подотрезка максимальной суммы в отсортированном массиве при условии, что максимальный элемент не больше суммы двух минимальных.
Алгоритм. Пусть длина такого отрезка >= 2 (один элемент очевидный случай). Переберем начало такого отрезка i, пусть $E = E_i + E_{i+1}$, найдем самого последнего игрока, у которого эффективность меньше либо равна E (юудем искать его бинарным поиском за $O(log(n))$), пусть его индекс $last$. Тогда самый длинный отрезок начинающийся с i это $[i; last]$. И мы можем включить в оптимальную команду всех этих игроков. Их суммарную эффективность будем вычислять используя предварительно подсчитанный массив частичных сумм за O(1). И каждый раз обновлять оптимальный ответ текущим.
Сложность. Сортировка элементов $O(nlog(n))$. Далее для каждого начала отрезка делаем бинарный поиск правого конца этого отрезка и за $O(1)$ находим сумму элементов этого отрезка. Итого вторая часть тоже $O(nlog(n))$.
Память. Расход памяти $O(n)$ (хотя это может зависеть от используемого алгоритма сортировки).