Формулировка задачи:

Задано конечное множество артефактов A = {a1,…,an} и размеры каждого артефакта s(a):

RAMi ∈ (0Гб, 32Гб], i ∈ А;

HDDi ∈ (0Гб, 500Гб), i ∈ А;

ОСi ∈ [Windows, Linux], i ∈ А;

*(Размерность артефакта и контейнера подходит под определение «Размерность пространства»)*

требуется найти такое разбиение множества А на непересекающиеся подмножества А1, …, Аk, чтобы сумма размеров артефактов в каждом подмножестве Аi не превосходила заданное ограничение конкретного контейнера и чтобы k было наименьшим возможным.

Можно считать, что артефакты , принадлежащие каждому множеству Аi , упаковываются в контейнеры разного размера, а цель состоит в упаковке артефактов множества U в как можно меньшее число контейнеров.

|  |  |
| --- | --- |
| Между узлами графа нет связей    Sn – сервер, Аn - артефакт | Хромосома:  S1.data = List[А1, А2]  S2.data = List[А3, А4]  S3.data = List[А5]  Ген: S1.data = List[А1, А2] или другой |
| 2 варианта первоначального заполнения:   1. При первоначальном заполнении хромосом учитываем ограничения на CPU\_Type, Level и ОС и не учитываем ограничения на HDD и RAM – выбрал этот вариант. 2. Заполняем хромосомы рандомом (не учитываем ограничения на CPU\_Type, Level, ОС, HDD и RAM). | |
| Правило скрещивания – берем список сервисов, расположенных на первом родителе. Этот список такой же, как и на втором родителе.  Пробегаемся по сервисам, четный сервис располагаем на ребенке на том же сервере, на котором он расположен на первом родителе. А на втором ребенке располагаем сервис на том сервере, на котором он расположен на втором родителе.  Нечетный сервис располагаем на ребенке на том же сервере, на котором он расположен на втором родителе. А на втором ребенке располагаем сервис на том сервере, на котором он расположен на первом родителе.  При таком скрещивании возможен выход за пределы памяти и др. метрик. | |
| Скрещивание: | |
| Функция приспособленности: В GeneticSharp всегда ищется максимум фитнес-функции  Возвращает оценку каждой хромосомы. Наиболее подходящее решение с наиболее высокой оценкой. Цель – снижение количества задействованных серверов. За несоответствие заданным ограничения присваиваем веса. По окончании анализа хромосомы масштабируем оценку приспособленности к выбранному диапазону. Если решение не имеет выхода за пределы ограничений, то оценка 0.  В ходе алгоритма решения с оценкой 0 записываются. По окончании алгоритма из коллекции выбирается решение с использованием наименьшего количества серверов. | |
| Мутация: с некоторой вероятностью берется случайный сервис из случайного сервера и перемещается на другой случайный сервер | |

Linux

Для достижения цели выпускной квалификационнонеобходимо выполнить следующие задачи: Для достижения цели выпускной квалификационной работы необходимо выполнить следующие задачи:

Windows

Для достижения цели выпускной квалификационной работы необходимо выполнить следующие задачи: Для достижения цели выпускной квалификационной работы необходимо выполнить следующие задачи:

Сервис 4

Сервис 3

Сервис 2

Сервис 1

GET

JSON-структура

[

{

“Name” : “Server\_1”,

“OS” : “Windows”,

“HDD\_full” : 250,

“HDD\_free” : 20,

“RAM\_full” : 20,

“RAM\_free” : 4,

“Services” : [

{

“Name” : “Microservice\_1,

“OS” : “Windows”,

“HDD” : 0.2,

“RAM” : 1,

},

{описание второго сервиса на сервере}

]

},

{описание второго сервера}

]

JSON-файл содержит информацию о сервере и коллекцию запущенных на нем сервисов.

Метод POST записывает структуру, оптимизированную алгоритмом, обратно в симулятор.