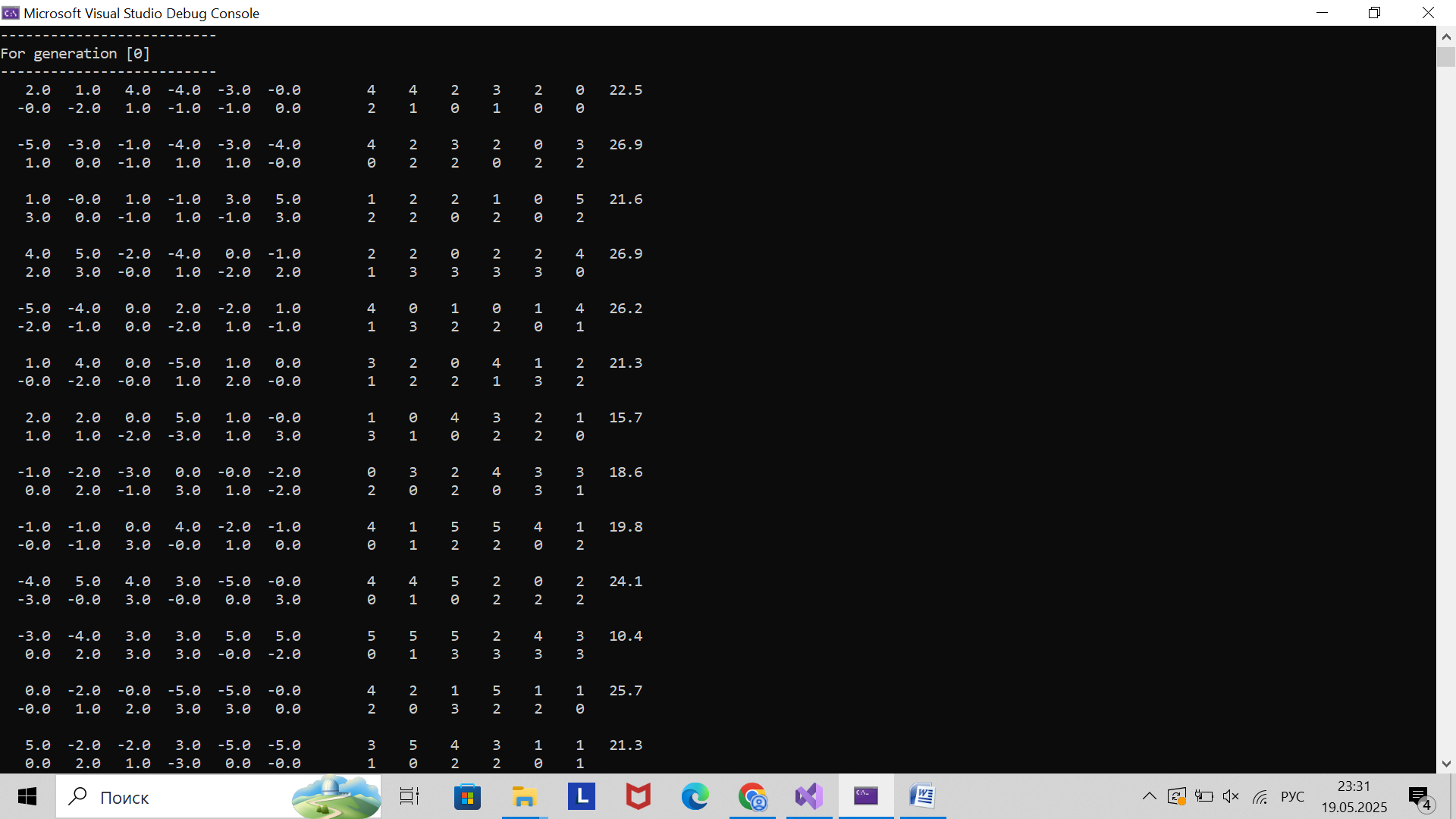
Отчёт

Код проекта можно увидеть на гитхабе по следующей ссылке:

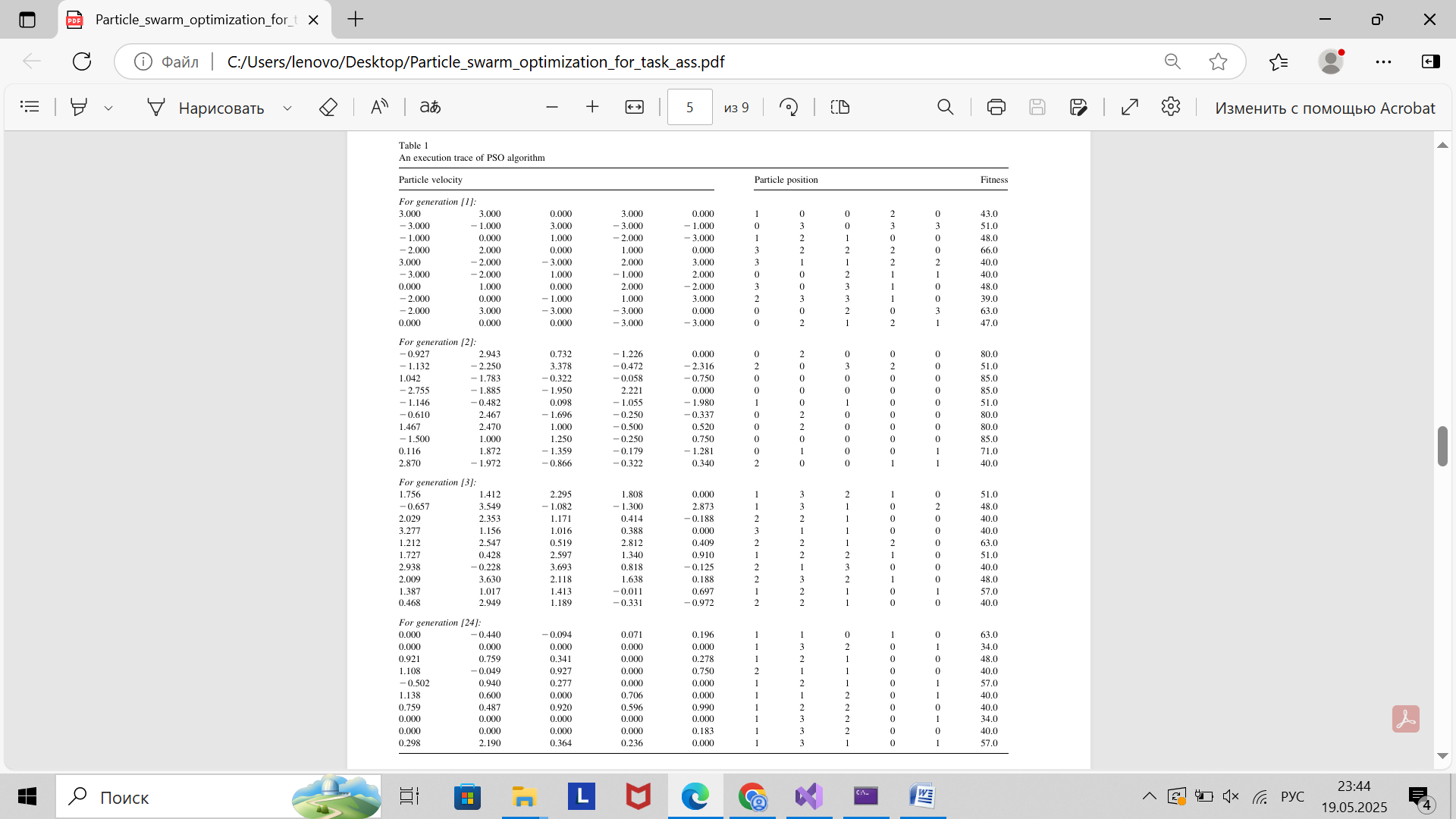
https://github.com/Njkita/TransportPSO/tree/main

Алгоритм PSO реализован внутри функции main. Модуль Astar, посвящённый выявлению коллизий и формированию путей, по которым будут двигаться агенты, реализован внутри хедера AstarPSOfinal.h и имеет удобную обёртку в виде функции run\_simulation, принимающей на вход файл с описанием графа и файл с описанием назначенных задач (сопоставлением номеров агентов, шахт и терминалов), по итогам своей работы записывающей в отдельный файл времена, затрачиваемые агентами на полный путь. Функция run\_simulation вызывается внутри фитнес-функции основного алгоритма main файла.

По итогам тестирования можно сказать, что алгоритм после преобразования работает корректно. В качестве примера приведу скрин с экрана, на котором можно видеть нулевое поколение частиц:

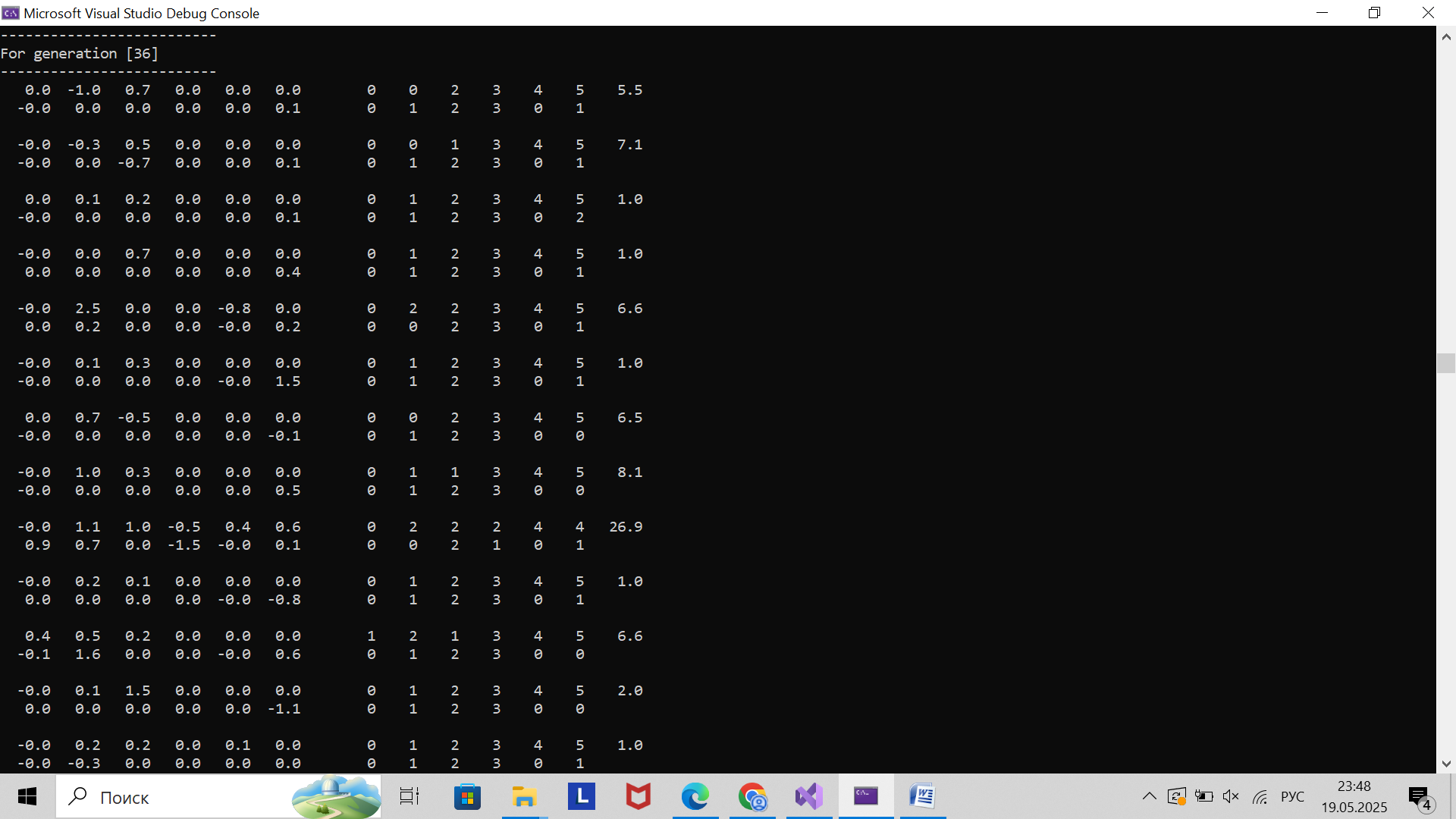


Здесь каждая частица имеет представление в виде двух строк: в первой содержится информация о терминалах (номера шахт не выписаны, они соответствуют индексам, то есть номерам, под которыми выписаны терминалы), а во второй – информация об агентах, число справа от каждой первой строчки – значение, возвращаемое фитнес-функцией по итогам анализа координат частицы. Шесть левых столбцов – многомерная скорость частицы, следующие шесть столбцов – координаты частицы (то есть назначенные задачи, сопоставления номеров шахт, терминалов и агентов). Вывод оформлен по аналогии со статьёй, на которую делался главный упор при разработке алгоритма:

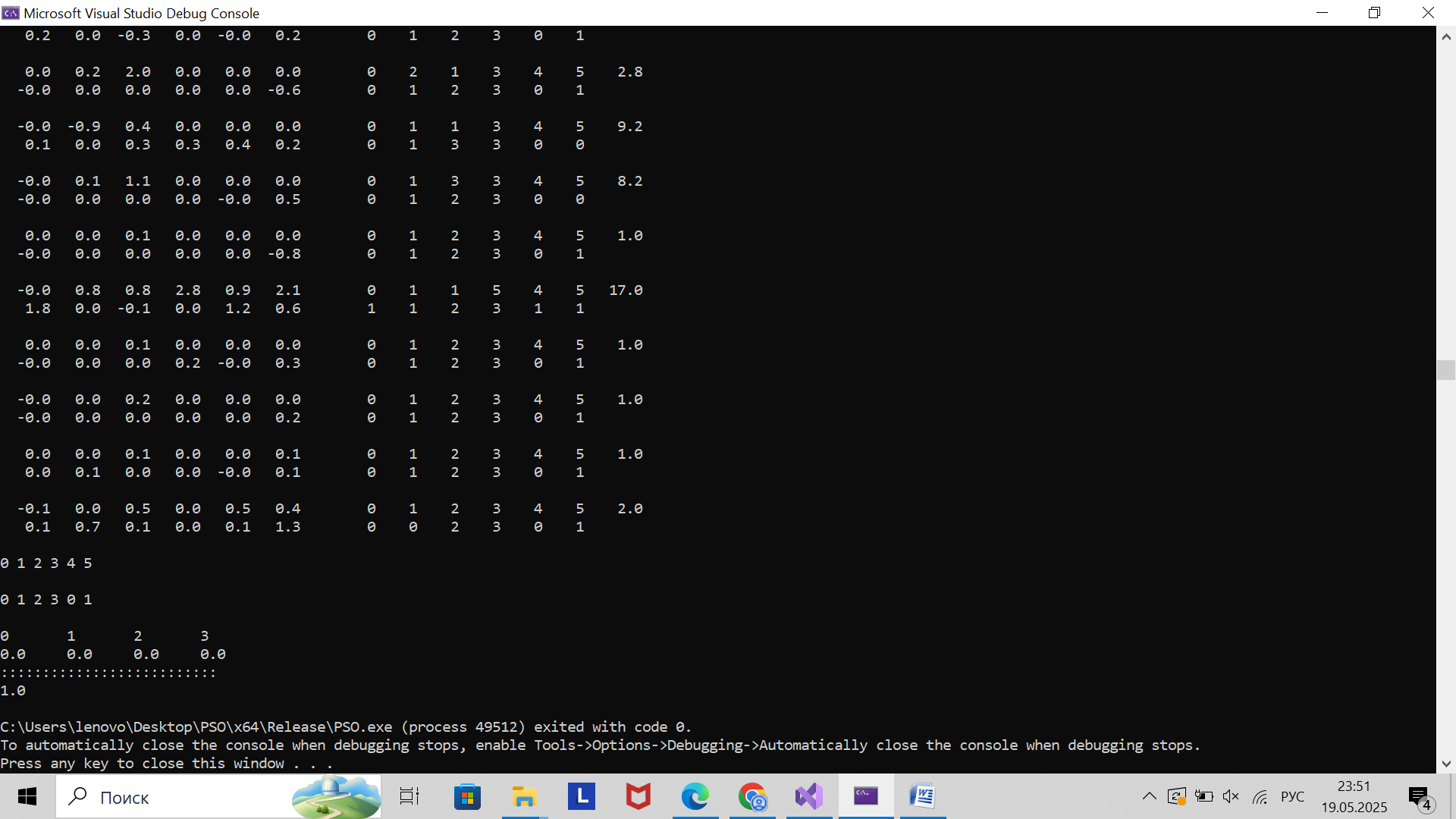


(Статья осталась сохранена).

Далее можно видеть неполное описание последнего поколения популяции частиц:



Здесь нужно обратить внимание на частицы со значением метрики, равным 1.0 – это частицы с оптимальным назначением, а также на их скорости, близкие к нулю, что свидетельствует о сходимости. Финальный результат с назначенными результатами можно видеть в конце вывода:



Номера шахт не выводятся – предполагается, что их номера идут по порядку. Соответствующие им терминалы выписаны в первой строчке (то есть здесь нулевой шахте соответствует нулевой терминал, первой шахте – первый терминал и т. д., что соответствует соответствию производства каждой из шахт и потреблению каждого терминала). Номера назначенных для перевозки груза из шахты в терминал агентов выписаны во второй строке (в данном случае используется четыре агента).

В данном примере алгоритм оптимально назначил терминалы и шахты, а так же равномерно распределил задачи между агентами.