Р.О. Мельник, магістр, гр. ПІ-46м, ФІКТ
Науковий керівник – д.т.н., проф. А.В. Панішев
Житомирський державний технологічний університет

РЕАЛІЗАЦІЯ МЕТОДУ РОЮ ЧАСТИНОК

Метод рою частинок спочатку призначався для імітації соціальної поведінки. Був розроблений для графічного моделювання зграї птиць. Кожна частинка має певне місце розташування і швидкість в просторі пошуку і, таким чином, характеризує певне рішення. Подібно до птахів, що переміщаються в навколишньому середовищі в пошуках їжі або при ухиленні від хижаків, частинки пролітають через простір пошуку, щоб знайти високоякісні рішення. В подальшому, він був розвинений для різних прикладних задач. В даний час, алгоритми рою застосовуються при вирішенні завдань чисельної і комбінаторної оптимізації, навчанні штучних нейронних мереж, побудові нечітких контролерів та в різних областях науки техніки:

- управлінні енергетичними системами;
- рішенням NP-важких комбінаторних проблем;
- завданні календарного планування;
- оптимізації;
- обробки зображень;
- розпізнаванні образів;
- кластеризації даних;
- проектуванні складних технічних систем.

Інтелект рою частинок можна назвати системою із множини агентів, яка володіє поведінкою самостійної організації. Метод рою частинок, в свою чергу, оптимізує функцію, підтримуючи популяцію можливих рішень, які називаються частинками, і переміщує їх в просторі, згідно заданої формули. Таке переміщення виконується по принципу найкращого знайденого в цьому просторі положення, яке змінюється, при знаходженні частинками більш вигідних положень. При тому, що не існує централізованої системи управління поведінкою, яка б вказувала кожній з частинок на те, що їй слід робити. Локальні і, в деякій мірі, випадкові взаємодії призводять до виникнення інтелектуальної глобальної поведінки.

При рішенні такої задачі, нам потрібно мати справу з оптимізацією функції, яка називається цільовою функцією, в залежності від багатьох параметрів. Оптимізація являє собою процес знаходження екстремуму (глобального максимуму або мінімуму) певної функції або вибору найкращого (оптимального) варіанту з безлічі можливих.

Алгоритм рою частинок являє собою n-мірне середовище, тобто область пошуку, в якій знаходяться агенти даного алгоритму (частинки). Із самого початку роботи алгоритму, ці частинки випадково розкидані по всій області пошуку, і кожна з цих частинок має свій випадковий вектор швидкості. В кожній точці, де побувала частинка, відбувається розрахунок значення цільової функції, при чому кожна з частинок запам’ятовує яке краще значення цільової функції вона знайшла, а також частинка знає, де знаходиться точка, яка є кращою серед усіх точок, які розвідали частинки. На кожній ітерації, не важливо чи виконання алгоритму йде через 1, 10, 100, 1000 ітерацій, частинки корегують свою швидкість, а саме модуль та направлення. Така ситуація відбувається для того, щоб з однієї сторони бути ближче до кращої точки, яку частинка знайшла сама через кожну задану ітерацію, та в той же час, приблизитись до точки, яка в даний момент являється глобально кращою. В залежності від кількості частинок та заданої кількості ітерацій, частинки мають зібратися якомога ближче до найбільш кращої точки, але може бути ситуація, коли деяка кількість частинок залишиться десь у відносно непоганому локальному екстремумі. Але самим головним в цьому є те, що хоча б одна із частинок знаходилась в околі глобального екстремуму.

Однією з особливостей даної реалізації цього методу є корекція швидкості. Корекція швидкості (1):

$$v_{i,t+1} = v_{i,t} + \varphi_p r_p (p_i - x_{i,t}) + \varphi_g r_g (g_i - x_{i,t}), \quad (1)$$

де $v_{i,t}$ – і-та компоненту швидкості при t операції алгоритму, $x_{i,t}$ – і-та координата частинки при t операції алгоритму, p_i - і-та координата кращого рішення, знайдена частинкою, g_i - і-та координата кращого рішення, знайдена всіма частинками, r_p, r_g – випадкові числа в заданому інтервалі (0, 1), φ_p, φ_g – вагові коефіцієнти, які потрібно вибирати під конкретну задачу. В нашому випадку це коефіцієнти власного кращого значення та глобального кращого значення відповідно.

Корегування поточної координати кожної частинки (2):

$$x_{i,t+1} = x_{i,t} + v_{i,t+1} \quad (2)$$

Після цього розраховуємо значення цільової функції в кожній новій точці, кожна частинка перевіряє чи стала нова координата кращою серед всіх точок, де вона побувала. Потім серед всіх нових точок відбувається перевірка, чи не знайшли ми нову глобальну кращу точку, і якщо це відбулось – запам’ятовуємо її координати і значення цільової функції в ній.

у (3):

$$\varphi = \varphi_p + \varphi_g, \quad (3)$$

де φ_p – власне краще значення, φ_g – глобальне краще значення. Канонічність алгоритму полягає в нормуванні коефіцієнтів власного кращого значення та глобального кращого значення, при умові, що $\varphi > 4$.

Основним недоліком даного алгоритму, як і в інших стохастичних алгоритмах оптимізації – не гарантована збіжність алгоритму при кінцевому числу частинок – звідси і збільшення кількості обчислень цільової функції.

Одним із плюсів алгоритму – зрозумілість та можливість швидко перебудувати реалізацію алгоритму на іншу формулу\метод для рішення алгоритму. В нашому випадку – використання функції Растрігіна, функція Швефеля та функція сфери.

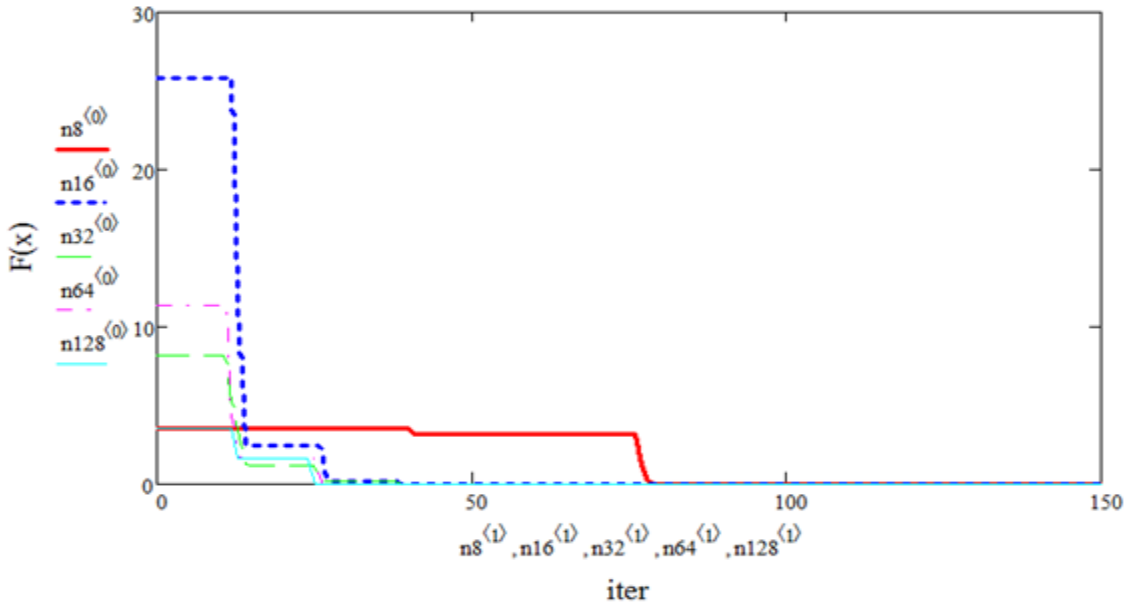

Рис. 1.

На рисунку 1 зображений результат роботи методу рою частинок з використанням сферичної функції в залежності від кількості частинок в рої для двовимірного простору.

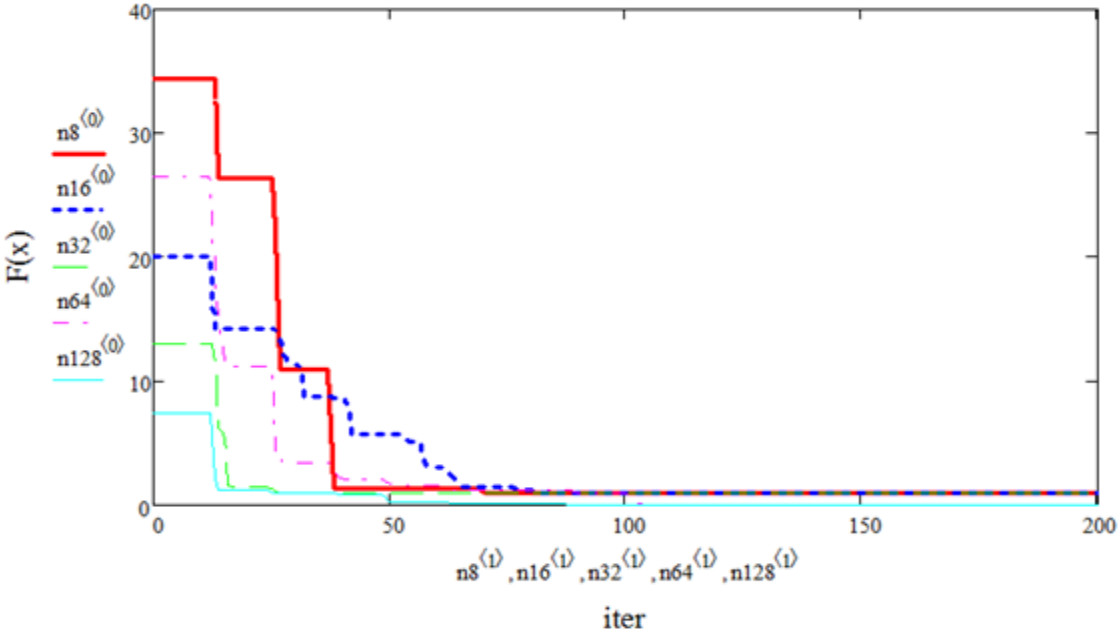

Рис. 2.

На рисунку 2 зображений результат роботи методу рою частинок з використанням функції Растрігіна в залежності від кількості частинок в рої.

Також під час роботи, був виконаний порівняльний аналіз між генетичними алгоритмами та роєвими алгоритмами