

# SGCS

## Документация

### Въведение

Как малки по размер организми като мравките и пчелите успяват да изпълняват сложни задачи, нужни за тяхното оцеляване? Координират се и с общия усилия компенсират малкия си размер и липсата на индивидуални възможности. В роботиката такива колаборативни системи имат няколко предимства пред по-сложни, но самостоятелно действащи роботи:

1. Всяко от тези роботчета е лесно и евтино за производство, заради по-малкото компоненти
2. По-малкият брой компоненти във всяко роботче също намалява риска от неизправност
3. Дори и няколко от роботчетата да изпитат неизправност, системата ще успее да изпълни задачата си

Има множество вече създадени малки роботчета, именно с цел да се използват групово. Едно от най-големите предизвикателства при разработването на такава система е координацията между индивидите. С SGCS се опитвам да предоставя приложимо решение на този проблем.

Мравките и пчелите превъзхождат повечето животински видове с възможността си да изследват околната среда, обменяйки по между си информация. Именно това ме вдъхнови да създам този проект. Като използвах колективното поведение на мравките за образец, създадох алгоритъм за координация между множество роботчета, който се нуждае само от комуникация между отделните индивиди на ограничено разстояние и вътрешна памет.

### Приложения

Такъв вид рояк от малки роботчета има множество практически приложения:

1. Изкуствено опрашване
2. Търсене на изчезнали хора в трудни условия
3. Търсене на неизбухнали мини и бомби след война, без това да застрашава човешки живот
4. Обследване на територии с ядрено, химично или биологично замърсяване
5. Наблюдение на стада и издирване на изгубили се животни
6. Наблюдение на частни домове и промишлени зони
7. Обследване на трудно достъпни терени

## Цел

Проектът има за цел да представи прост алгоритъм, който всеки робот в колаборативна система да следва, за наблюдение и претърсване на площ. Предоставя се симулация, която демонстрира алгоритъма и позволява настройката на множество параметри.

## Съществуващи решения

Множество системи за координация на рояк от малки роботчета съществуват, но всяка една от тях има недостатъци, които аз опитвам да разреша.

### Karma

*Karma* е система, която се базира на модела кошер-дрон. За успешната си работа се изисква постоянна комуникация на всяко роботче с централен компютър, което е практически невъзможно, тъй като най-голямото предизвикателство при направата на миниатюрни роботчета е източникът на енергия, който трябва да е достатъчно лек, за да може роботчето да се движи, но да съдържа достатъчно енергия за продължително движение. Това от своя страна лимитира обхвата на комуникация. Освен това, в случай че централният компютър се повреди, цялата система не може да функционира, което може да се окаже фатално при някои приложения.

### Phormica

*Phormica* е друга система за организация на рояк от малки роботчета. В тази система роботчетата имат възможността да оставят изкуствени феромони след себе си, като прожектират UV светлина върху пода, който е покрит с фотохромен материал. Тази система не е подходяща поради 2 основни причини:

1. Изисква всяко роботче да има компоненти за прожектиране на UV светлина и за последващо засичане къде е била прожектирана такава светлина
2. Изисква покритие с фотохромен материал, което е скъпо и означава, че хора трябва да обходят цялата площ преди роякът да бъде пуснат

## Алгоритъм

Моят алгоритъм използва виртуални феромонни карти. Всяка роботче пази в паметта си къде поставя феромони и използва тази карта при вземането на решения. Когато две роботчета са в комуникационен обхват, обменят

феромонните си карти. По този начин системата не зависи от централен компютър и успява да функционира дори и при лимитиран обсег на комуникация. Не се изискват никакви специални хардуерни компоненти.

Когато роботчето решава накъде да се движи, на случаен принцип се избират няколко възможни посоки (броят може да се промени от параметрите на симулацията). За всяка от тези възможни посоки се изчислява привлекателност на

бъдещата позиция  $\sum_{i=1}^n d_i \times s_i$ , където  $n$  е броят на феромоните,  $d_i$  е разстоянието до  $i$ -тия феромон и  $s_i$  е силата му. По този начин се избира позицията, която е най-вероятно да доведе до необходим участък. Освен това всяка позиция се проверява дали позволява на робота да се върне до началната точка с останалата батерия. Ако нито една бъдеща позиция не изпълнява това условие, то роботчето се връща до началната точка и там се презарежда.

## Имплементация

Симулацията е написана на Java, а за графичния интерфейс съм използвал фреймуърк JavaFX. Това позволява симулацията да работи на всички операционни системи с лесен и интуитивен интерфейс. Построяването на проекта се автоматизира с Maven. С помощта на JLink и Jpackage се генерира лесно за инсталиране приложение за всяка операционна система и по този начин симулацията може да бъде използвана и на компютър без инсталирани Java и JavaFX.

## Програмна структура

Програмата се състои от 7 класа:

SGCS:

Това е главният клас на програмата, който настройва JavaFX прозореца и стартира интерфейса.

SimulationControl:

Този клас е *Singleton*, отговорен за интерфейса, параметрите и контрола на симулацията.

## ShowTrailCheckboxes:

Това е помощен клас на **SimulationControl**, който е изцяло отговорен за бутоните, които задават чии феромони да се показват на екрана. Също представлява *Singleton*.

## Simulation:

Този *Singleton* клас е отговорен за симулацията и координира всички обекти участващи в нея.

## Loop:

**Loop** е *Singleton* клас, отговорен за анимацията с JavaFX.

## Bot:

Това е клас, който представлява едно роботче.

## Pheromone:

Това е клас, представляващ един феромон.

# Инсталация

Хипервръзката към проекта води до GitHub хранилището на кода. От там натискате бутона Releases, откъдето можете да свалите .msi инсталатор на програмата. След като го стартирате, програмата автоматично ще се инсталира под името SGCS и ще се създаде shortcut на десктопа Ви.

# Използване

## Параметри

Следните параметри могат да бъдат зададени на симулацията:

### Брой работи:

Броят на симулираните роботчета. Може да е между 1 и 1000, но за симулацията на голям брой работи се изискват много компютърни ресурси.

Обсег на комуникация:

На какво разстояние два робота могат да обменят феромонните си карти. Може да е между 1 и 10000.

Шанс за повреда:

Какъв е шансът в проценти всяка секунда роботът да спре да работи.

Скорост:

С каква скорост се движи всеки робот. Може да е от 0 до 10.

Разреждане на батерията:

Колко процента на секунда се разрежда батерията на всеки робот.

Преглеждани бъдещи позиции:

Измежду колко посоки избира роботът при вземането на решение. Може да е от 1-1000, но при голям брой се изискват много компютърни ресурси.

Бързина на отслабване на феромоните:

С колко процента на секунда отслабва всеки феромон.

След като параметрите са зададени, симулацията може да бъде стартирана с бутона *Start Simulation*.

След като симулацията е стартирана, зададените параметри ще бъдат изписани на екрана, а под тях може да се изберат феромоните на кои работи да се показват на екрана. Чрез бутонът *Pause/Play* симулацията може да бъде спирана и пускана.

Програмата може да бъде затворена по всяко време с бутона *Quit*.