

**Универзитет у Нишу**  
**Електронски факултет Ниш**

**Интелигентни системи**  
**Други домаћи задатак – *Swarm* алгоритми**

Студент:  
Димитрије Јовић, 928

Ниш, децембар 2019. год.

## С А Д Р Ж А Ј

1. Формулација проблема .....	3
2. Имплементација <i>Swarm</i> алгоритма.....	4
3. Референце.....	6

## 1. ФОРМУЛАЦИЈА ПРОБЛЕМА

Наш јунак из претходне приче, Мирко, непосредно пре опкладе са својим оцем, се запослио као достављач новина у локалној штампарији. С обзиром да се достављање новина врши у раним јутарњим часовима, Мирко је имао великих проблема. Наиме, Мирко је скоро сваког дана морао да устаје рано, да оде до штампарије и да достави новине на већ предвиђене локације, након тога би ишао у школу, у коју је због посла увек каснио. Како је сваког дана каснио, и деловао све уморније, за његово здравље се забринуо његов друг из клупе, Миладин, ког смо у претходној причи упознали као врсног програмера. Испричавши своју јутарњу рутину, Миладин је обећао Мирку да ће му помоћи. Миладин је већ сутрадан позвао Мирка и показао му своје решење. Наиме, Миладин је креирао апликацију која врши оптимизацију руте којом Мирко свако јутро доставља новине, тако да се достава врши најкраћом могућом рутом. Апликација је базирана на биолошки инспирисаним алгоритмима, односно алгоритмима који рекреирају понашање појединих животиња. Мирко је у почетку био неповерљив, јер није знао ништа у вези тих алгоритама, међутим, када је у пракси почео да се држи руте коју је направила апликација, приметио је да се његово време доставе новина, као и сама ефикасност доставе, знатно побољшали.

Newspaper delivery

Best initial trail length: 112.0

New best length of 111.0 found at iteration 2

New best length of 101.0 found at iteration 5

New best length of 78.0 found at iteration 9

New best length of 72.0 found at iteration 10

New best length of 65.0 found at iteration 11

New best length of 61.0 found at iteration 12

New best length of 48.0 found at iteration 13

New best length of 44.0 found at iteration 18

New best length of 43.0 found at iteration 22

New best length of 42.0 found at iteration 23

New best length of 41.0 found at iteration 26

New best length of 40.0 found at iteration 58

New best length of 39.0 found at iteration 73

New best length of 38.0 found at iteration 217

**Main parameters**

Number of places:

30

Number of ants:

7

Number of iterations:

10000

**Influence of adjacent node distance**

Parameter alpha:

3

Parameter beta:

2

**Pheromone factors**

Increase factor:

2.0

Decrease factor:

0.01

**Best initial route**

15->17->24->3->18->19->1->5->28->11->22->0->2->14->9->26->25->16->7->8->4->23->29->20->6->21->13->27->10->12

**Best found route**

5->4->23->0->3->16->15->25->19->17->2->7->6->9->27->13->12->28->8->22->21->20->10->29->1->18->26->11->24->14

Find

Resume

Pause

Stop

Reset

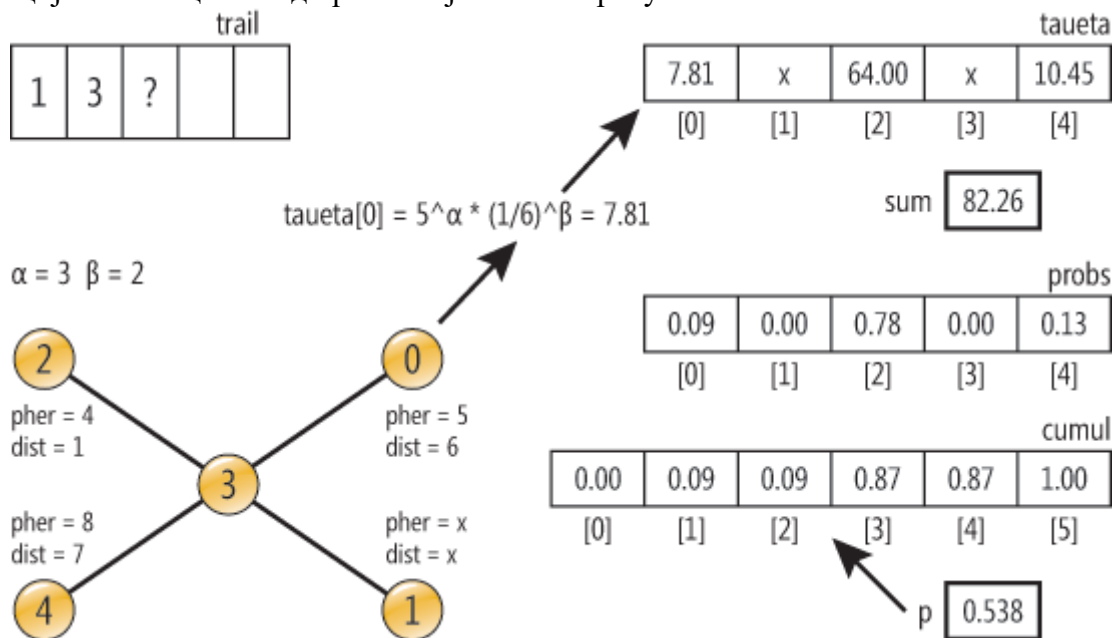
Status: Finished

## 2. ИМПЛЕМЕНТАЦИЈА SWARM АЛГОРИТМА

*ACO* алгоритам је техника вештачке интелигенције, заснована на понашању мрава код полагања феромона, може се користити за проналажење изузетно сложених проблема који траже оптималан пут кроз граф. Феромони су течност коју мрави остављају за собом и на тај начин привлаче друге мраве. Више мравна ће путовати краћим трагом до извора хране - и депонирати више феромона - него на дужим стазама. Феромони полако испаравају током времена.

*ACO* захтева спецификацију неколико параметара као што су фактор утицаја феромона, коефицијент испаравања феромона, коефицијент додавања феромона. Кључна идеја *ACO*-а је употреба симулираних феромона који привлаче мраве на боље путање кроз граф. Алгоритам врши наизменично ажурирање путања мравна на основу тренутних вредности феромона и ажурирање феромона на основу нових путања мравна. Четири најзначајнија елемента су: иницијализација мравна, иницијализација феромона, ажурирање мравна, ажурирање феромона. У конкретном случају, као решење се приказује оптимална путања, односно редослед оптималног обиласка локација.

У имплементацији *ACO* алгоритма, мрави су представљени у облику низа, где сваки мрав представља низ целих бројева. Ти бројеви представљају редослед обиласка локација. Кључ *ACO* алгоритма је поступак којим се ажурира сваки мрав односно путања тако што конструише нову, надамо се бољу, путању засновану на информацијама о феромону и удаљености. Главни проблем је одредити наредну локацију коју је потребно обићи. На основу растојања између суседних локација и јачине феромона између тих локација, врши се израчунавање вероватноће одабира локације. На слици испод приказан је начин израчунавања.



Сваки елемент *taueta* низа се израчунава по формули  $\text{taueta}[i] = \text{pher}_{i,j}^{\alpha} * (1/\text{dist}_{i,j})^{\beta}$ . Елементи низа *probs* се рачунају по формули  $\text{probs}[i] = \text{taueta}[i] / \sum \text{taueta}$ . Низ *cumul* служи за одабир локације која ће следећа бити посећена, тако што се генерише случајан број између 0 и 1, и провери се између која два елемента низа *cumul* упада. Сваки елемент се рачуна по формули  $\text{cumul}[i+1] = \text{probs}[i] + \text{cumul}[i]$ , где је  $\text{cumul}[0] = 0$ .

Информације о феромонима су смештене у симетричној матрици, у којој индекси врсти и колона представљају локације. Све вредности су у почетку постављене на неку произвољну малу вредност. Што се тиче ажурирања феромона, једноставније је него ажурирање мрава, односно путање. Свака вредност феромона се смањује, симулирајући испаравање, и повећава се, симулирајући таложење феромона на путањи. Смањења феромона се израчунава множењем тренутне вредности феромона фактором мањим од 1.0, што зависи од параметра испарења. Што је тај параметар већи, већи је и пад вредности феромона. Повећање феромона се израчунава додавањем удела укупне дужине путање тренутног мрава. Удео се одређује параметром повећања феромона, веће вредности параметра повећавају количину додатог феромона.

Треба напоменути да је коришћена готова имплементација алгоритма, и да је модификована ради уклапања у сам концепт апликације.

### 3. РЕФЕРЕНЦЕ

- [1] <https://docs.microsoft.com/en-us/archive/msdn-magazine/2012/february/test-run-ant-colony-optimization>
- [2] <https://docs.microsoft.com/en-us/archive/msdn-magazine/2012/february/february-2012-code-downloads>