

Системи, основани на знания – 2020/2021 учебна година

***Тема 4:
Генетични алгоритми***

Дефиниция на генетичен алгоритъм

Вариант на ***стохастично търсене в лъч***, при което новите състояния се генерират чрез комбиниране на двойки родителски състояния вместо чрез модифициране на текущото състояние.

Основни принципи

- Състоянията се представят като низове над дадена крайна азбука (често като низове от нули и единици).
- Оценяваща функция (fitness function): оценява пригодността (близостта до целта) на съответното състояние. Има по-големи стойности за по-добрите състояния.
- Алгоритъмът започва работа с множество (популация) от k случайно генерирани състояния (поколение 0).
- Принципи на получаването на състоянията от следващите поколения: селекция, кръстосване, мутация.

Селекция

В генерирането на състоянията от следващото поколение участват някои от най-добрите представители на текущото поколение (съгласно оценяващата функция), избрани на случаен принцип.

Кръстосване

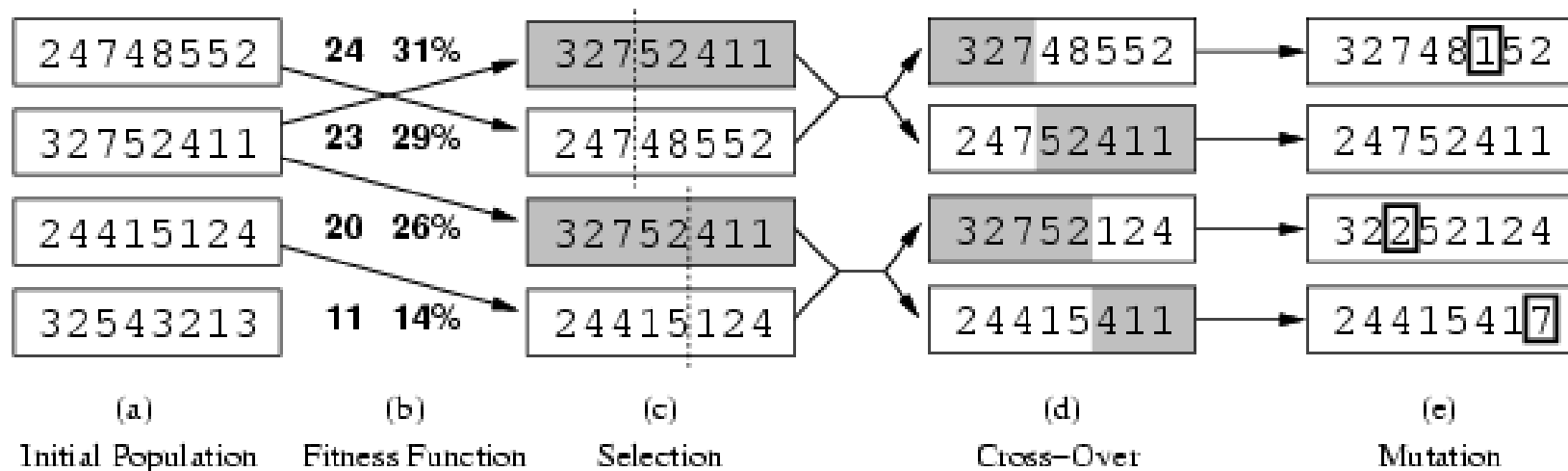
Избират се двойка „родителски“ състояния и се определя т. нар. *точка на кръстосването* им (позиция в двата низа). Състоянието – наследник се получава чрез конкатенация на началната част на първия и крайната част на втория родител. Възможно е да се получи и друг наследник, в конструирането на който участват неизползваните части на двамата родители.

Мутация

Извършване на случайни промени в случайно избрана малка част от новата популация с цел да се осигури възможност за достигане на всяка точка от пространството на състоянията и да се избегне опасността от попадане в локален екстремум.

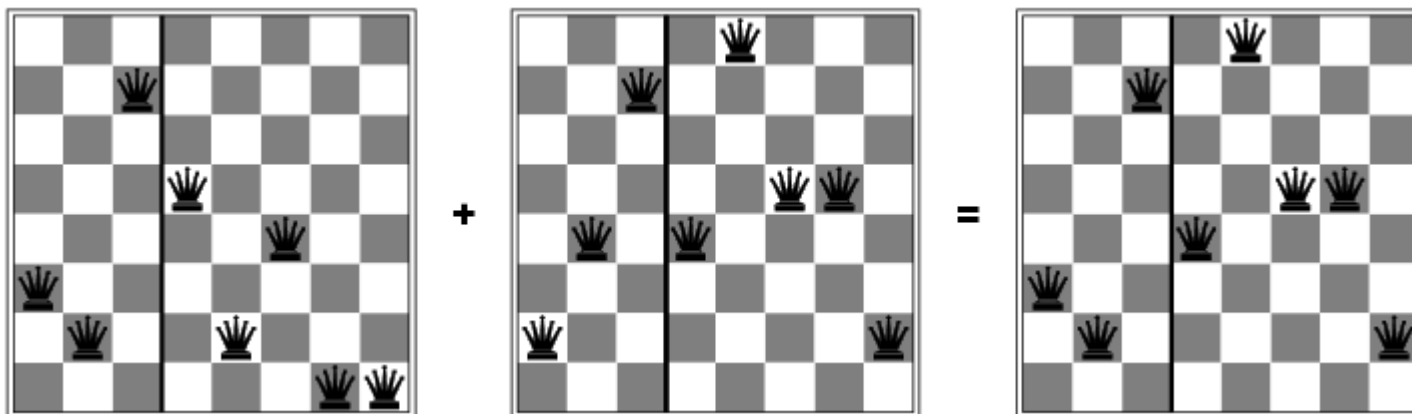
Пример: задача за 8-те царици. Да се намери такова разположение на 8 царици върху шахматна дъска, че никоя двойка от тях да не се бият взаимно.

- Представяне на състоянията: всяко състояние се представя чрез 8-компонентен вектор, i -тият компонент на който представя вертикалната позиция p_i на i -тата царица.



- Оценяваща функция: за всяко състояние връща като резултат броя на двойките царици, които не се бият взаимно ($\min = 0$, $\max = 8 \times 7/2 = 28$)
- $24/(24+23+20+11) = 31\%$
- $23/(24+23+20+11) = 29\%$ и т.н.

Пример за кръстосване



function GENETIC-ALGORITHM(*population*, FITNESS-FN) **returns** an individual

inputs: *population*, a set of individuals

FITNESS-FN, a function that measures the fitness of an individual

repeat

new_population \leftarrow empty set

loop for *i* **from** 1 **to** SIZE(*population*) **do**

x \leftarrow RANDOM-SELECTION(*population*, FITNESS-FN)

y \leftarrow RANDOM-SELECTION(*population*, FITNESS-FN)

child \leftarrow REPRODUCE(*x*, *y*)

if (small random probability) **then** *child* \leftarrow MUTATE(*child*)

add *child* **to** *new_population*

population \leftarrow *new_population*

until some individual is fit enough, or enough time has elapsed

return the best individual in *population*, according to FITNESS-FN

function REPRODUCE(*x*, *y*) **returns** an individual

inputs: *x*, *y*, parent individuals

n \leftarrow LENGTH(*x*)

c \leftarrow random number from 1 to *n*

return APPEND(SUBSTRING(*x*, 1, *c*), SUBSTRING(*y*, *c* + 1, *n*))

Примери за различни типове кръстосване

- **Кръстосване в единична точка**

Избира се една точка на кръстосване. Низът – резултат от началото си до точката на кръстосване е копие на началната част на единия родител, останалата му част е копие на съответната част на втория родител.

Parent 1		Parent 2		Offspring
8691247536	+	1234567892	=	8691567892

- **Кръстосване в две точки**

Избират се две точки на кръстосване. Низът – резултат от началото си до първата точка на кръстосване е копие на съответната част от първия родител, частта на резултата от първата точка на кръстосване до втората точка на кръстосване е копие на съответната част на втория родител и останалото е копие на оставащата след втората точка на кръстосване част на първия родител.

Parent 1		Parent 2		Offspring
8691247536	+	1234567892	=	8691567536

- **Аритметично кръстосване**

Извършва се определена операция (аритметична, логическа и т.н.) между двамата родители и в резултат се получава новото потомство.

$$\begin{array}{ccc} \text{Parent 1} & & \text{Parent 2} \\ 1101011010 & + & 1101111101 \\ \hline & = & 1101011000 \\ \text{Offspring} \end{array}$$

двоично кодиране, операция AND между съответните битове

Характерни приложения на генетичните алгоритми

- Решаване на оптимизационни задачи (задача за търговския пътник, задача за раницата и др.)
- Решаване на задачи за удовлетворяване на ограничения
- Избор на стратегия при игри за двама (по-общо, N) играчи
- Самообучение на невронни мрежи (уточняване на теглата на връзките между елементите в невронна мрежа с определена архитектура)
- Генетично програмиране

Пример: задача за търговския пътник. Дадени са група градове и разстоянията между тях. Търговският пътник трябва да посети всички градове, като минимизира общата дължина на пътя. Да се намери такъв маршрут на търговския пътник, който започва от даден град, завършва в същия град, включва всички останали градове точно по веднъж и образува път с възможно най-малка обща дължина.

Кодиране (представяне на състоянията)

Състоянията се представят като поредици от означенията на съответните градове в реда, в който търговският пътник ще ги посети.

Пример: състоянието, представено чрез низа “153264798”, означава маршрут на търговския пътник, съставен от следната поредица от градове (предполага се, че градовете са означени с цифрите от 1 до 9): 1-5-3-2-6-4-7-9-8-1.

Кръстосване

Може да бъде както в единична точка, така и в две точки. И в двата случая са необходими допълнителни действия, които да осигурят получаването на коректно състояние като резултат от кръстосването.

Пример: кръстосване в две точки с получаване на поколение от двама наследници.

Ако $p_1 = \text{"1234 | 567 | 8"}$ и
 $p_2 = \text{"8521 | 364 | 7"}$

са две родителски състояния, то при кръстосването им ще се получат

$c_1 = \text{"1234 | 364 | 8"}$ и
 $c_2 = \text{"8521 | 567 | 7"}$

Очевидно s_1 и s_2 представят некоректни състояния, затова се прибегва до следната допълнителна операция:
 размените $3 \leftrightarrow 5$, $6 \leftrightarrow 6$ и $4 \leftrightarrow 7$, които са извършени при
 кръстосването на родителските състояния в областта между двете
 точки на кръстосване, се повтарят в областите извън точките на
 кръстосване.

Така окончателно се получава потомство

$$d_1 = \text{"1257 | 364 | 8"} \quad \text{и}$$

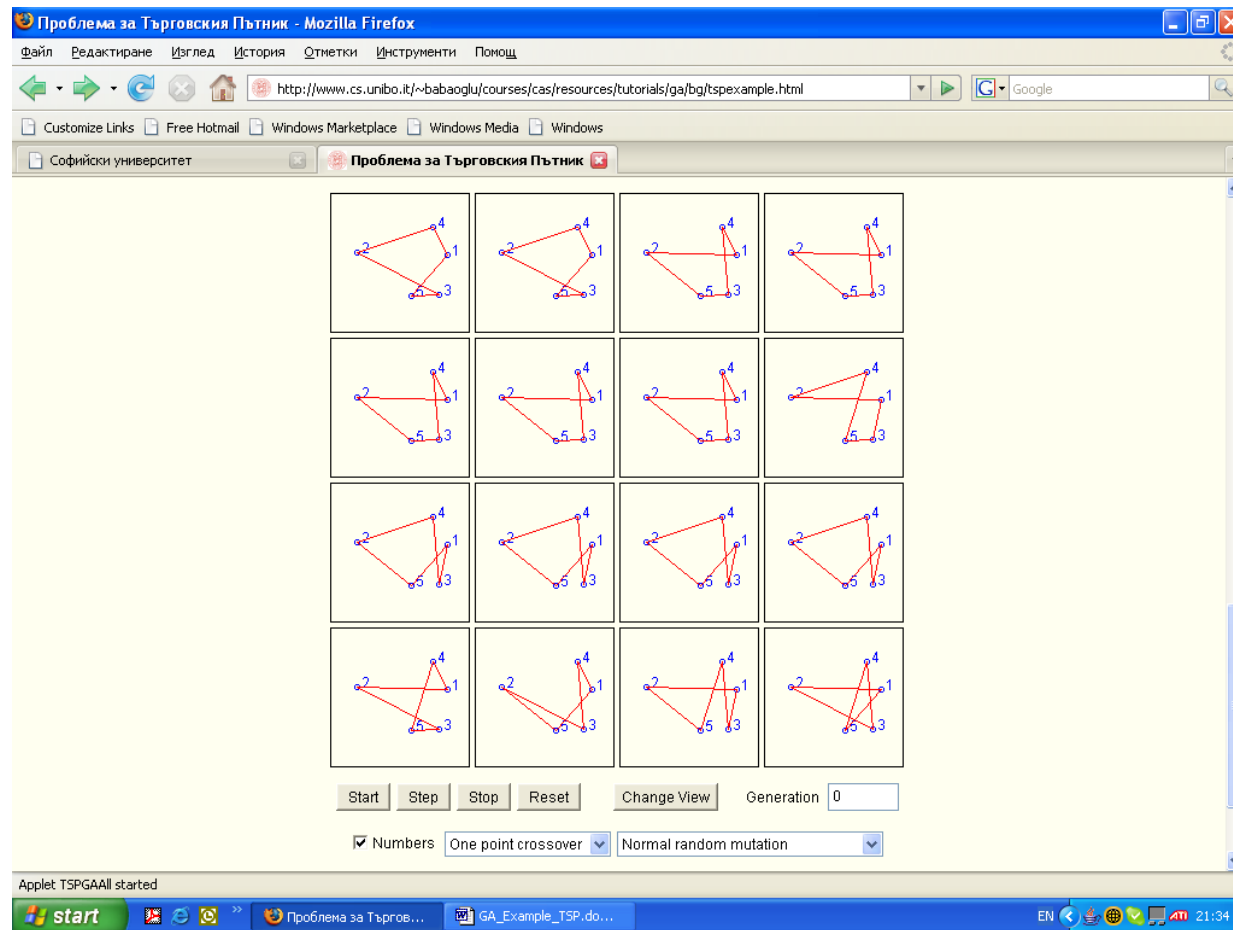
$$d_2 = \text{"8321 | 567 | 4"}$$

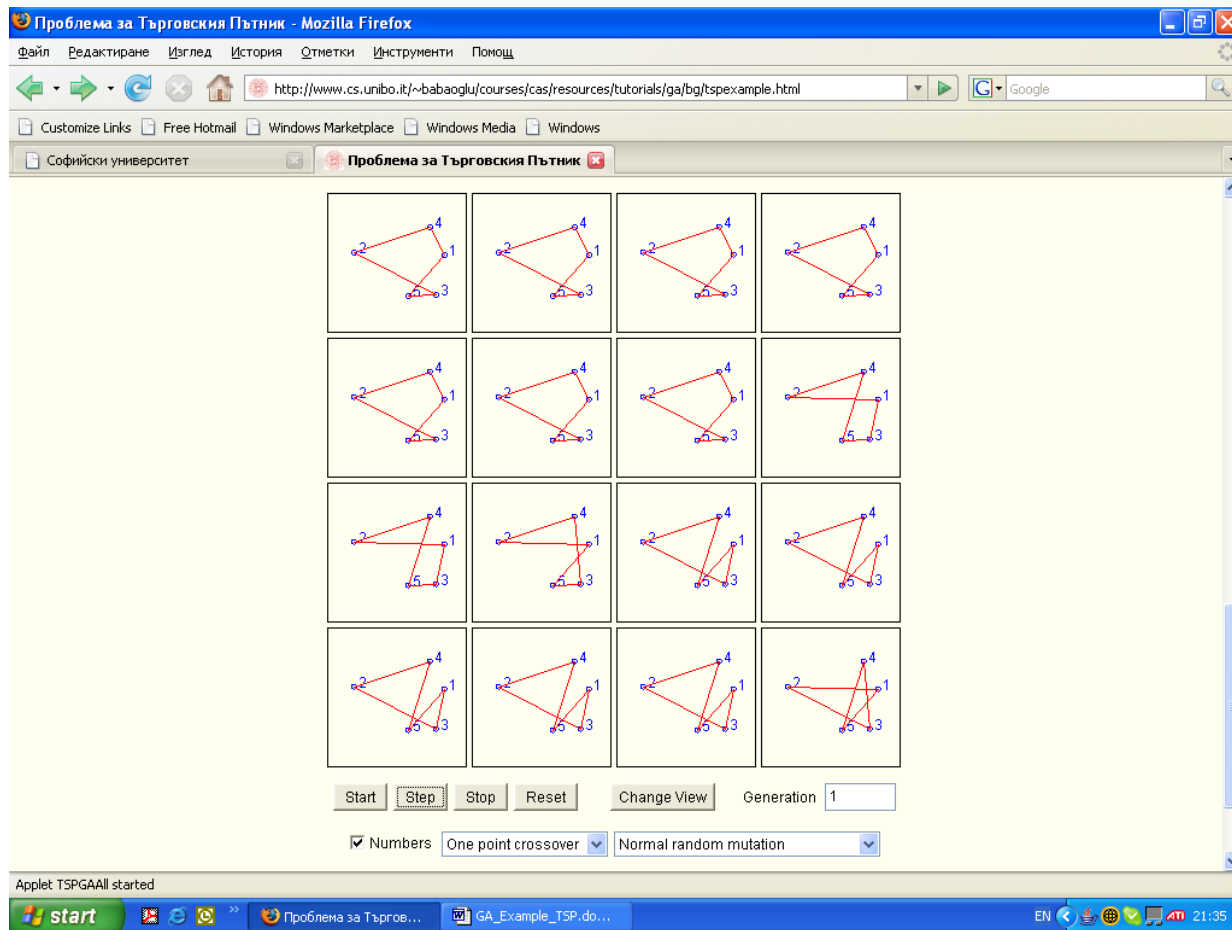
Мутация

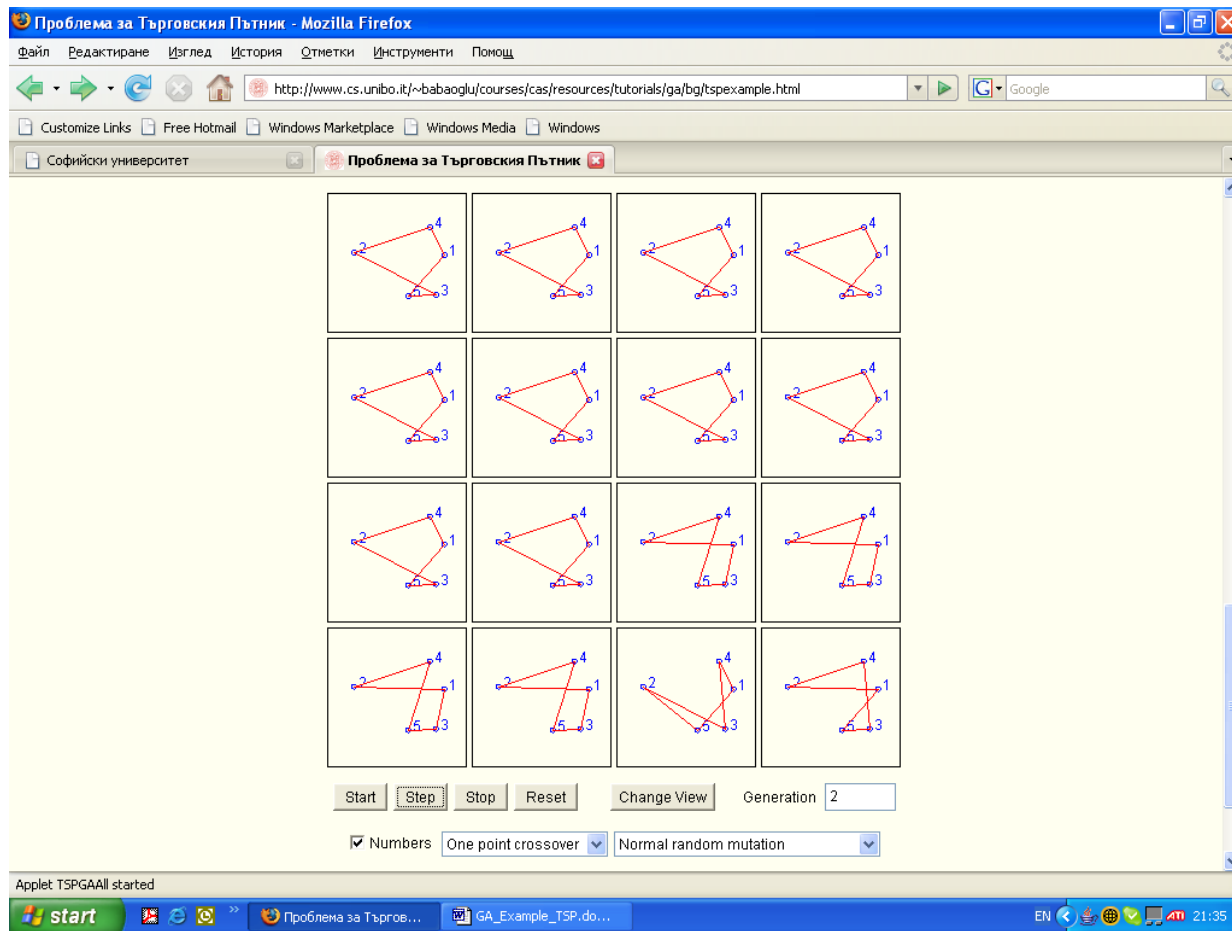
Размяна на два случайно избрани знака от съответния низ.

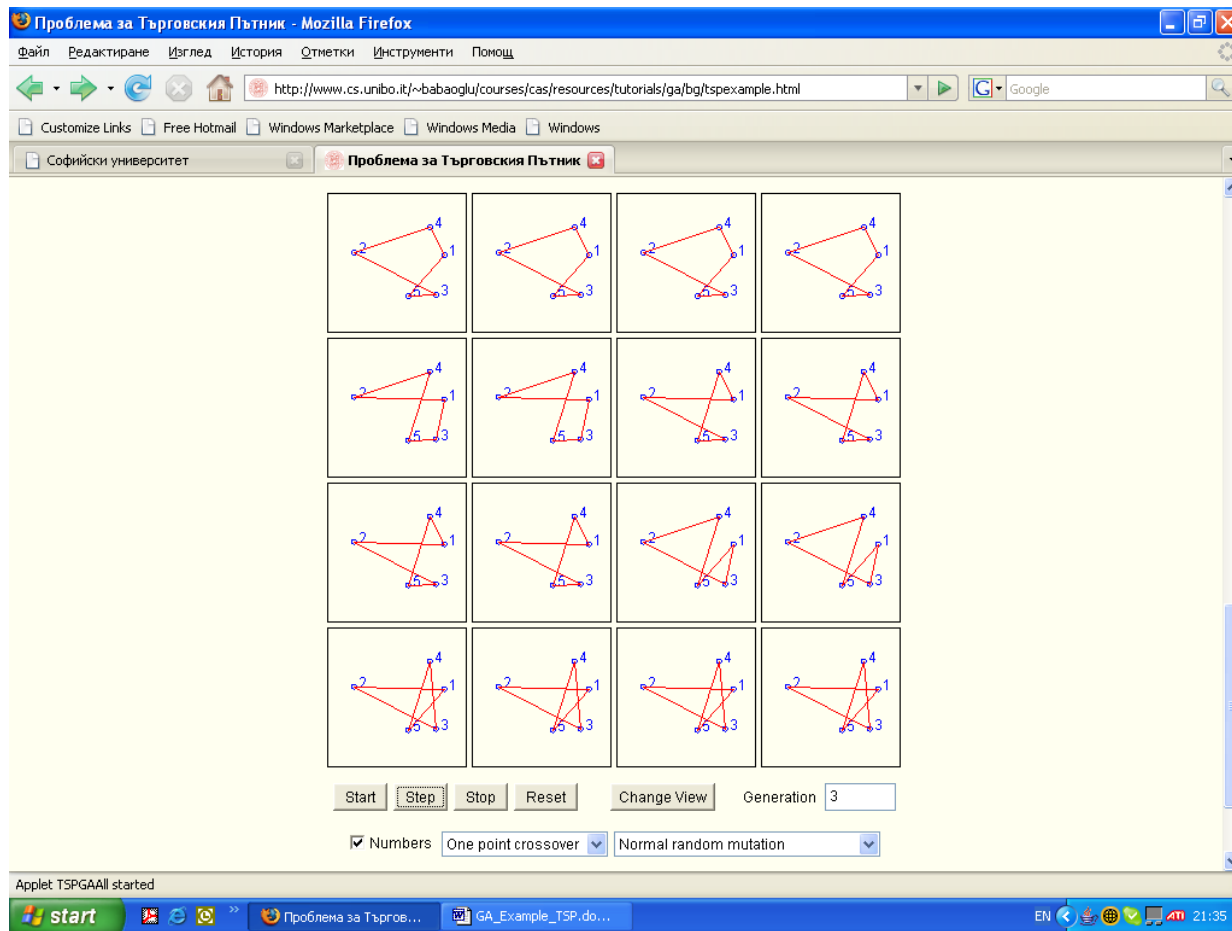
Пример: “153264798” → “153864792”

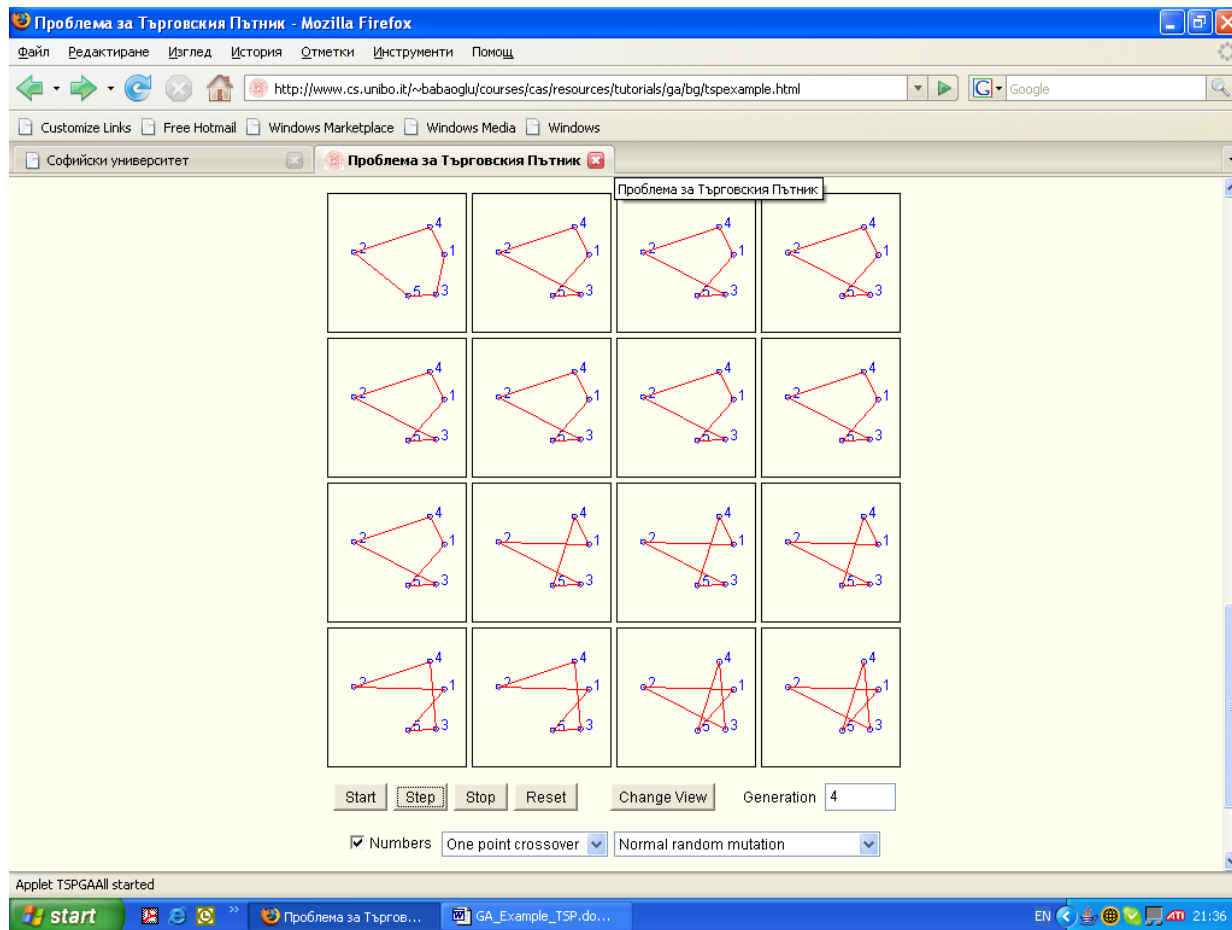
Проследяване на изпълнението на генетичен алгоритъм за решаване на задачата за търговския пътник:

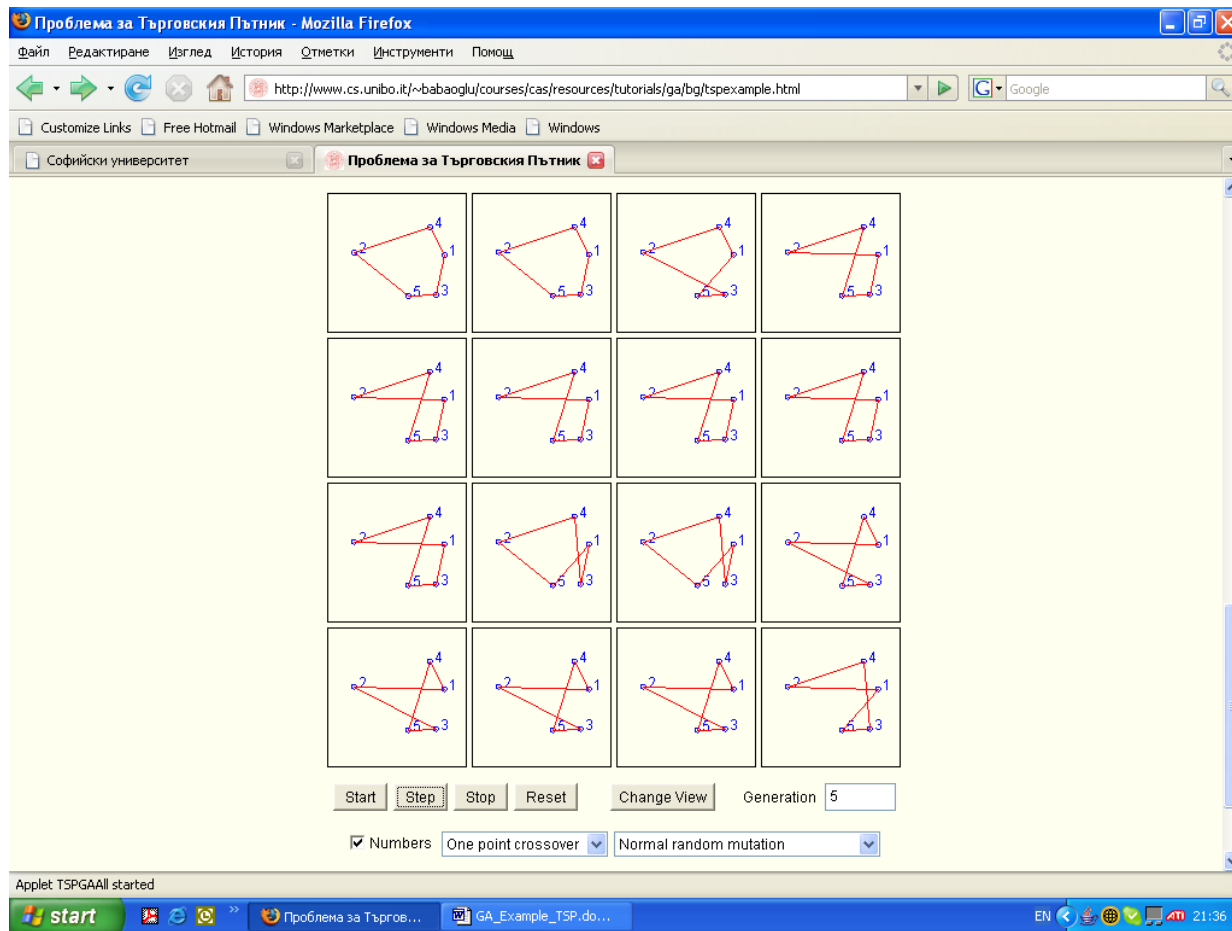












Прекратяване на изпълнението на даден генетичен алгоритъм

В зависимост от конкретната задача могат да бъдат използвани и други критерии (освен посочените в общата дефиниция) за проверка на „сходимостта“ и вземане на решение за прекратяване на изпълнението на един генетичен алгоритъм. Такива критерии, които често се използват на практика, са:

- съвпадение/близост на оценките на най-добрите състояния от последните няколко популации;
- наличие на висок процент състояния от текущата популация, които са идентични или подобни едно на друго.