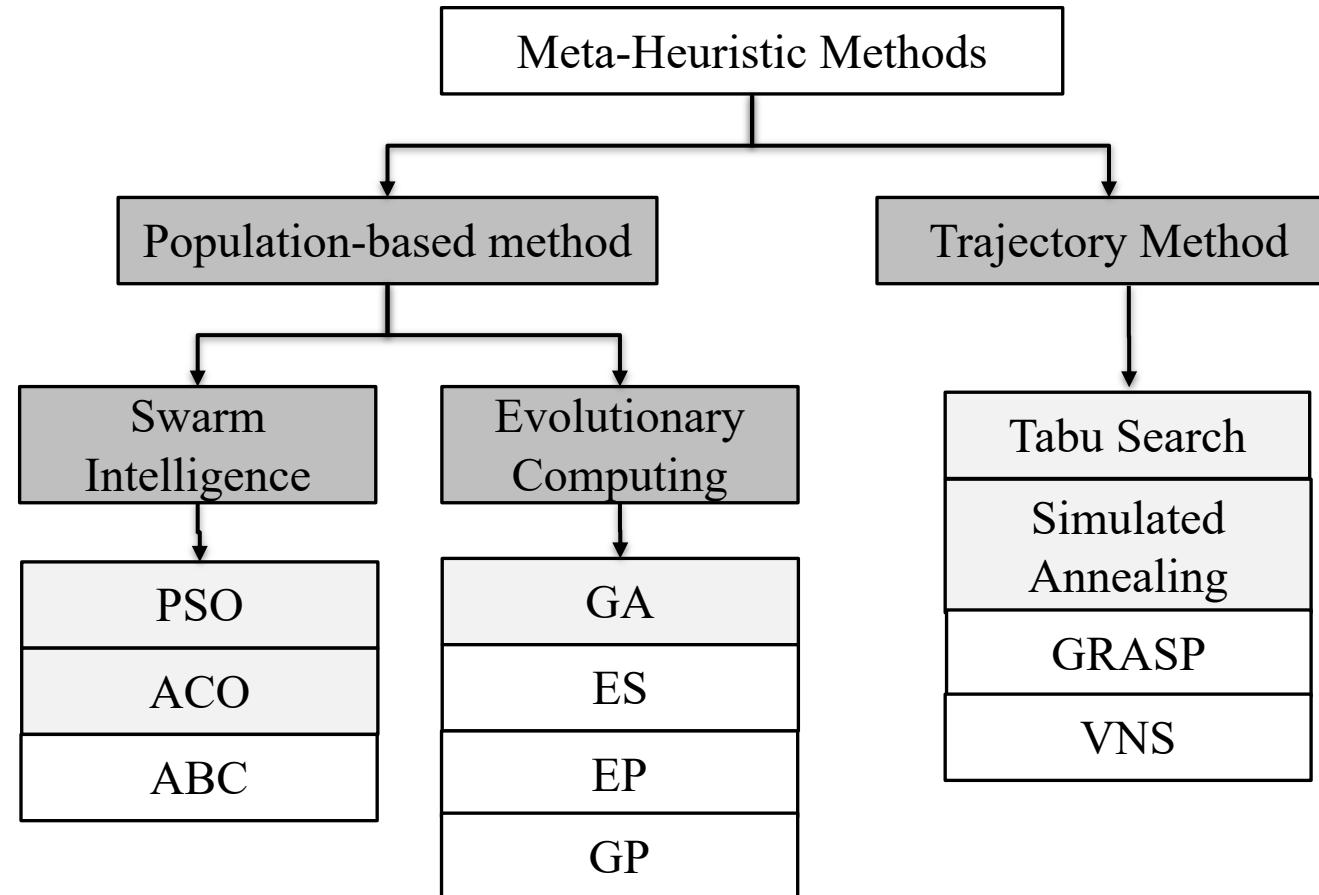


POPULATION BASED PROBLEM SOLVING



GA BASED OPTIMAL SENSOR PLACEMENT

META-HEURISTICS



Genetic Algorithms

Based on material and slides from A. E. Eiben and J. E. Smith. "Introduction to Evolutionary Computing". Springer, 2003.

|| GENETIC ALGORITHMS

Based on the Darwinian theory of evolution,

The theory offers an explanation of the biological diversity and its underlying mechanisms,

Natural selection plays an important role.

Natural selection favours those individuals that **compete** for the given resource most effectively, those who are **adapted** to fit the environmental conditions best,

This phenomenon is known as ***survival of the fittest***.

|| GENETIC ALGORITHMS

The fitness is determined by the *phenotypic traits*

- Behavioural features,
- Physical features.

Selected individuals reproduce, passing their properties to the offspring,

Other individuals die without mating, their properties are thus discarded.

GENETIC ALGORITHMS

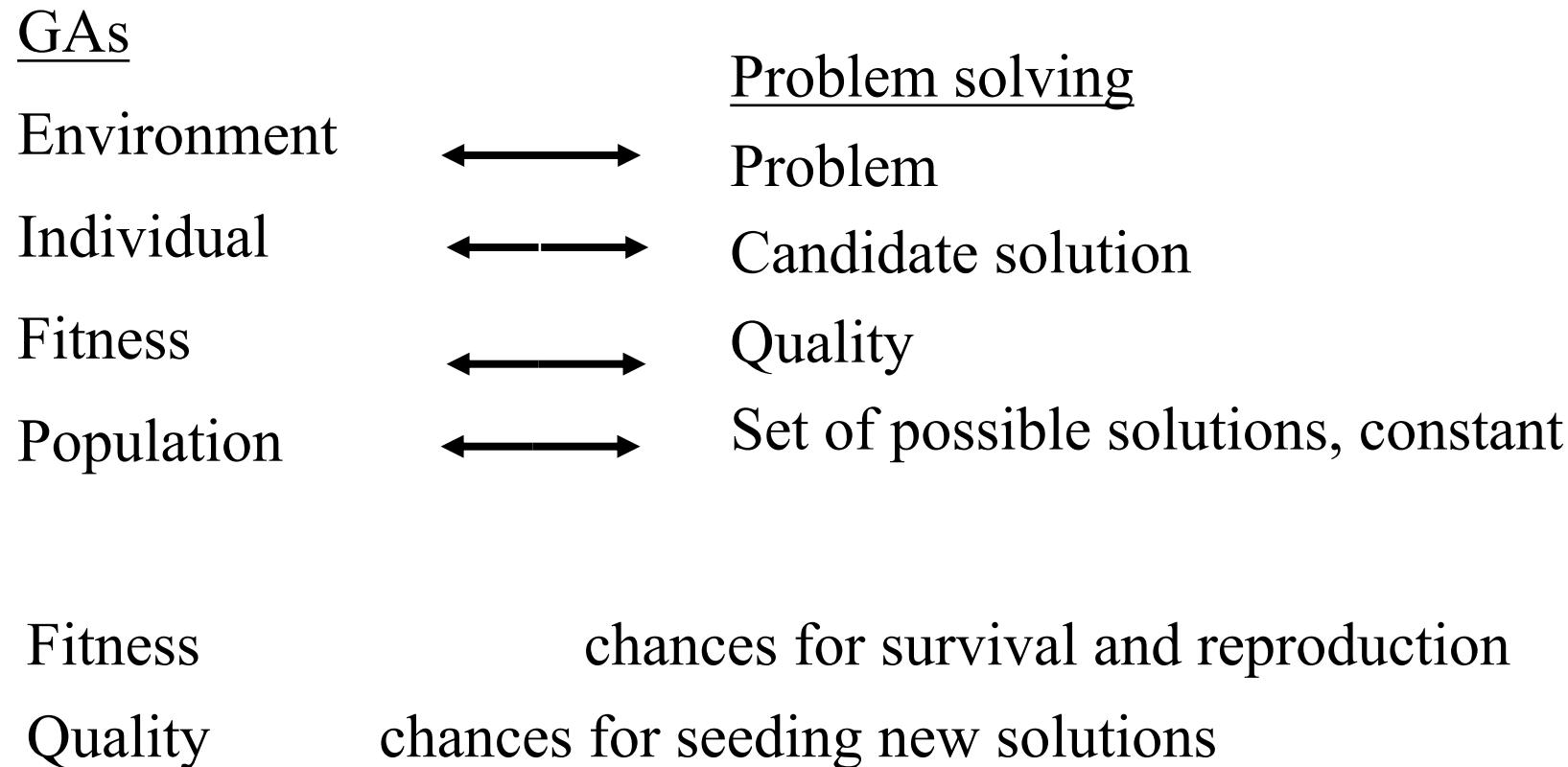
Darwin also recognized that small and random variations, ***mutations***, occur in the phenotypic traits during reproduction,

Through these variations, new combinations of traits occur and get evaluated,

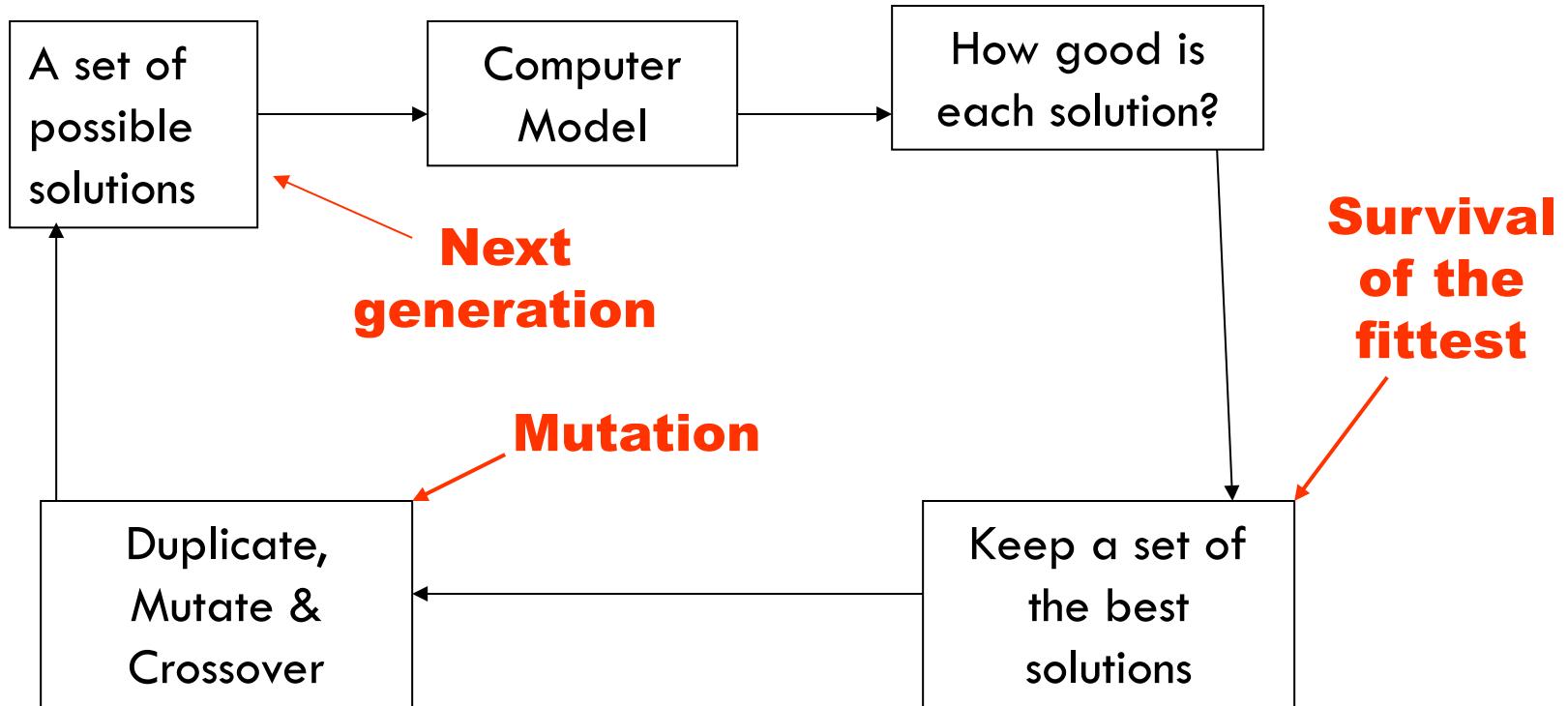
The basic operations are hence ***selection***, ***reproduction*** and ***mutation***.

|| GENETIC ALGORITHMS

- A genetic algorithm maintains a **population of candidate solutions** for the problem at hand, and makes it evolve by *iteratively applying a set of stochastic operators*



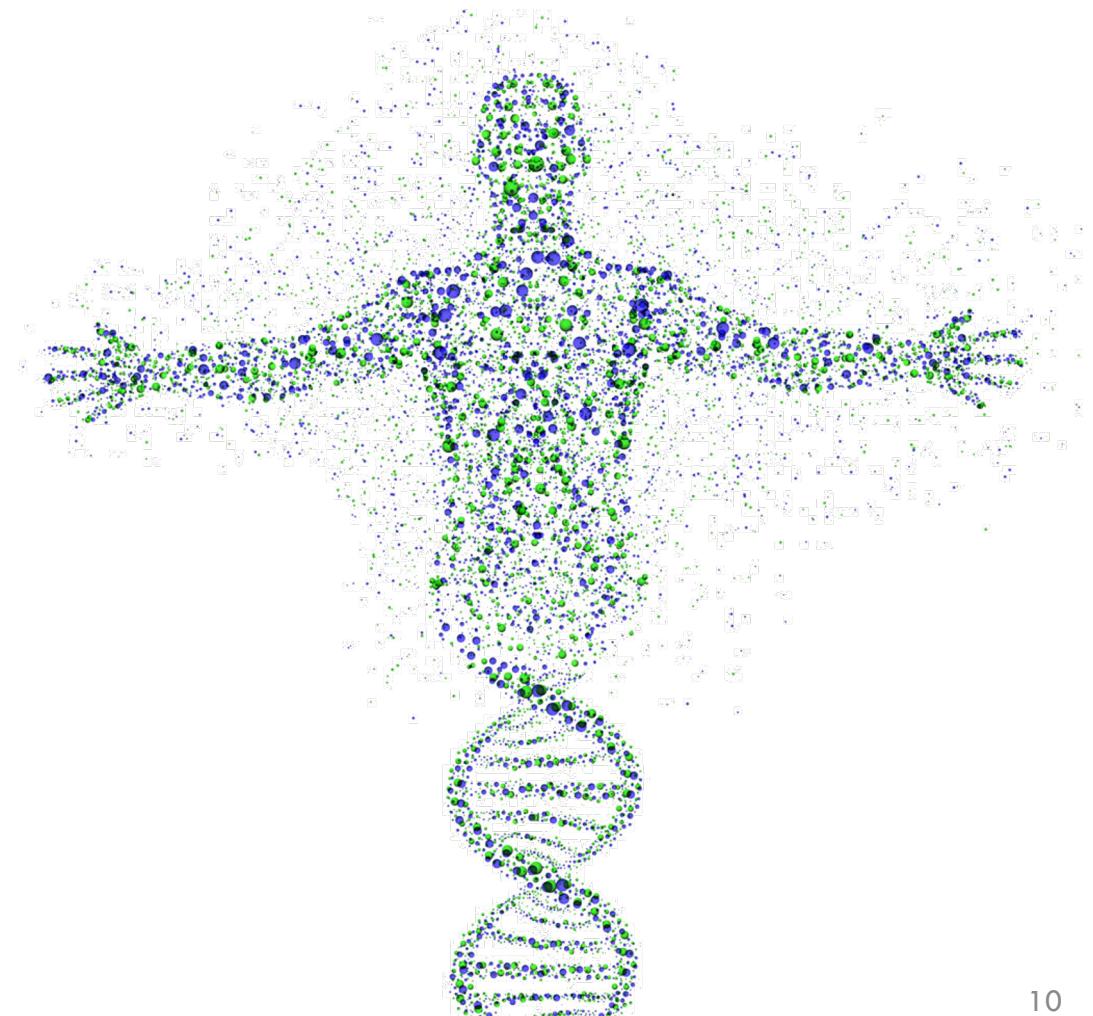
GENETIC ALGORITHMS



|| GENETIC ALGORITHMS

First GA was developed by Holland in 1975, referred to as **SGA** (Simple GA),

Sometimes referred to as the *classical* or *canonical* GA.



Simple Genetic Algorithms

Based on material and slides from A. E. Eiben and J. E. Smith. "Introduction to Evolutionary Computing". Springer, 2003.

|| SGA - INTRODUCTION

GAs handle a *population* of *individuals* (solutions),

The probability of selecting a bad solution is reduced by handling multiple good solutions,

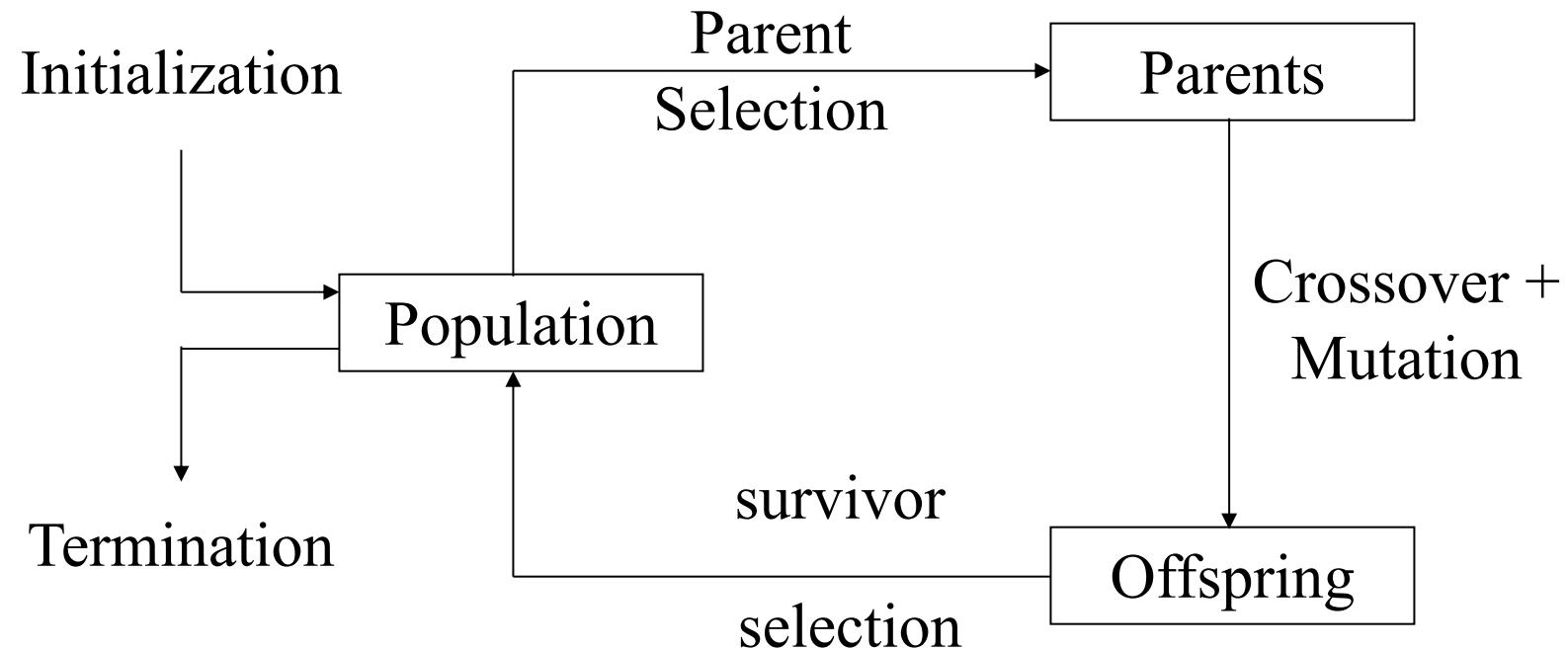
The population evolves from one iteration to the next according to selection and the ***search operators (genetic operators)***,

Search operators:

- Recombination (Crossover),
- Mutation.

|| SGA - INTRODUCTION

The general scheme:



SGA INGREDIENTS

Representation	Binary Strings
Recombination	1-point, N-point, or uniform
Mutation	Bitwise bit-flipping with fixed probability
Parent selection	Fitness-Proportionate
Survivor selection	All children replace parents
Speciality	Emphasis on crossover

SGA – THE ALGORITHM

Initialize population with random candidates,

Evaluate all individuals,

While *termination criteria* is not met

- Select parents,
- Apply crossover,
- Mutate offspring,
- Replace current generation,

end while

SGA – THE ALGORITHM

The *termination criteria* could be:

- A specified number of generations (or fitness evaluations),
- A minimum threshold reached,
- No improvement in the best individual for a specified number of generations,
- Memory/time constraints,
- Combinations of the above.

Simple Genetic Algorithms

Representation

SGA - REPRESENTATION

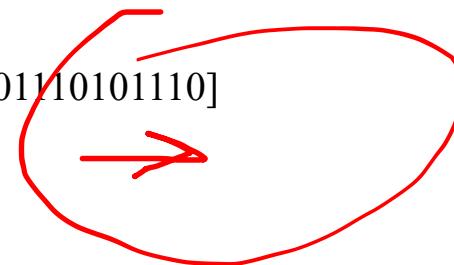
A mapping is applied from the parameter domain to the binary domain,

Phenotype, a vector in the parameter domain,

Abstract representation of Phenotype is called **chromosomes** or **genotype**

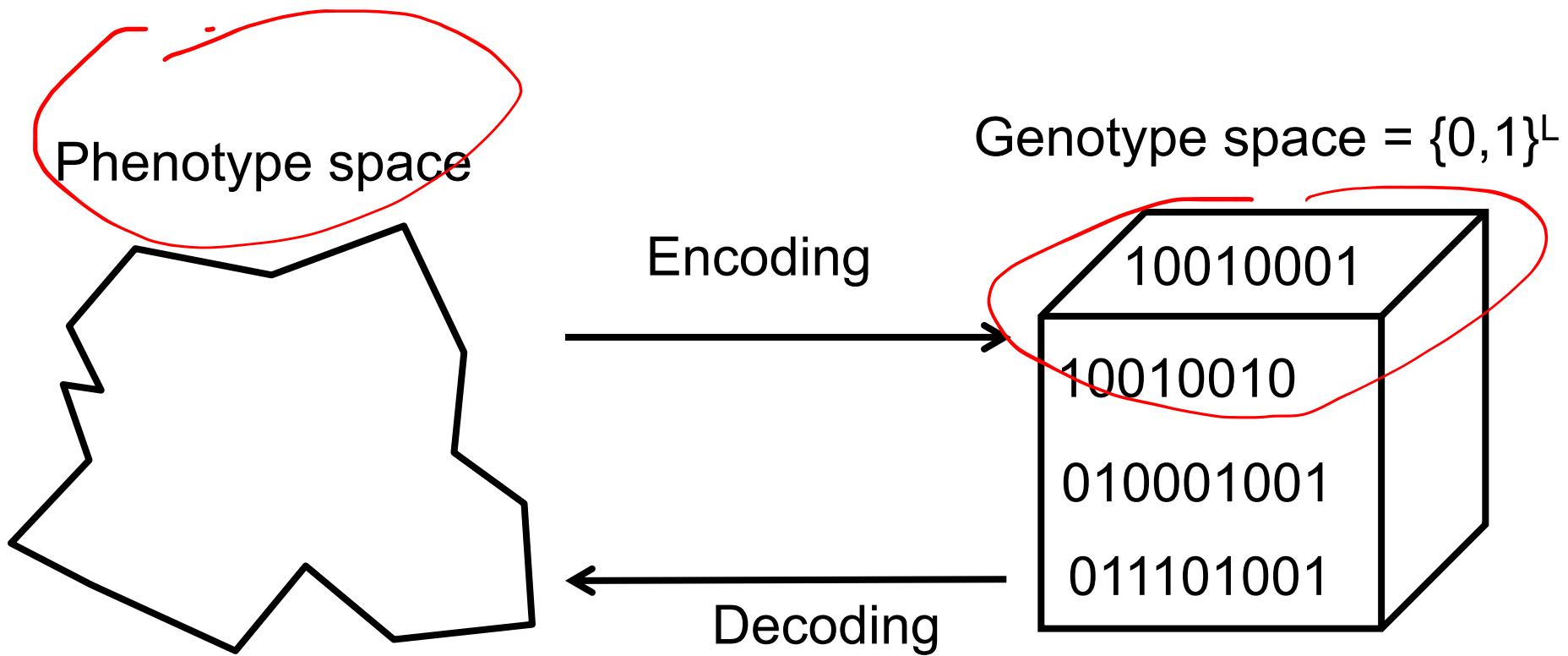
In SGA **Genotype**, is a string in the binary domain.

- 0 and 1 for the presence or absence of the features [Chromosomes: 001110101110]
- Order of genes on chromosome can be important



Good coding is probably the most important factor for the performance of a GA

SGA - REPRESENTATION



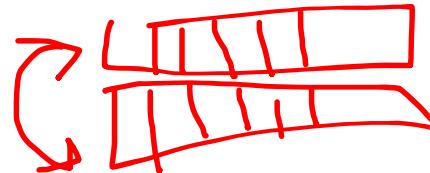
Simple Genetic Algorithms

Selection

SGA REPRODUCTION CYCLE

Select parents for the mating pool (size of mating pool = population size)

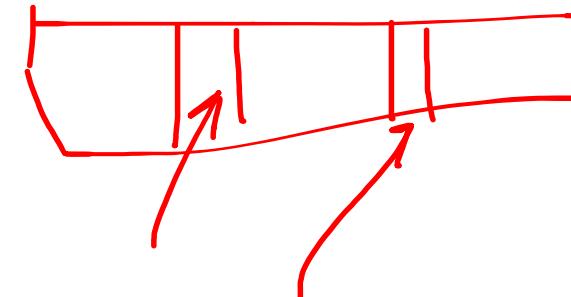
Shuffle the mating pool



Apply crossover for each consecutive pair with probability (p_c) otherwise copy parents

Apply mutation for each offspring (bit-flip with probability (p_m) independently for each bit)

Replace the whole population with the resulting offspring



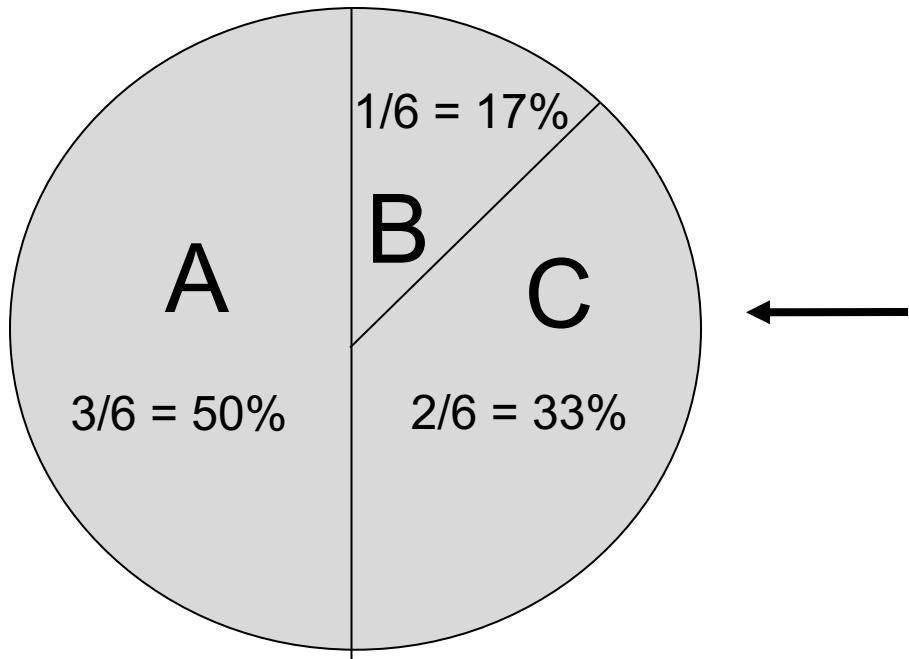
SGA - SELECTION

Parents are selected with a probability proportional to their fitness, ***proportional selection***.

Main idea: better individuals get higher chance:

- Chances proportional to fitness,
- Implementation: roulette wheel technique
 - Assign to each individual a part of the roulette wheel,
 - Spin the wheel n times to select n individuals.

SGA - SELECTION



$$\text{Fitness}(A) = 3$$

$$\text{Fitness}(B) = 1$$

$$\text{Fitness}(C) = 2$$

Simple Genetic Algorithms

Crossover

SGA - CROSSOVER

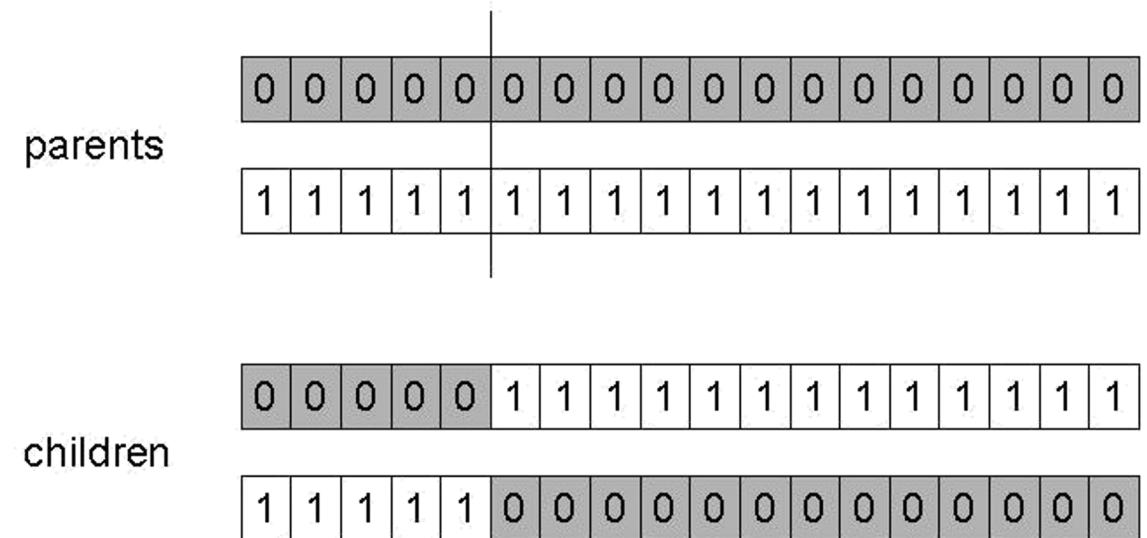
Crossover is applied according to a probability (P_c)

Otherwise, the two parent are copied to the offspring unmodified

P_c typically in range (0.6, 0.9)

1-point crossover (the binary case)

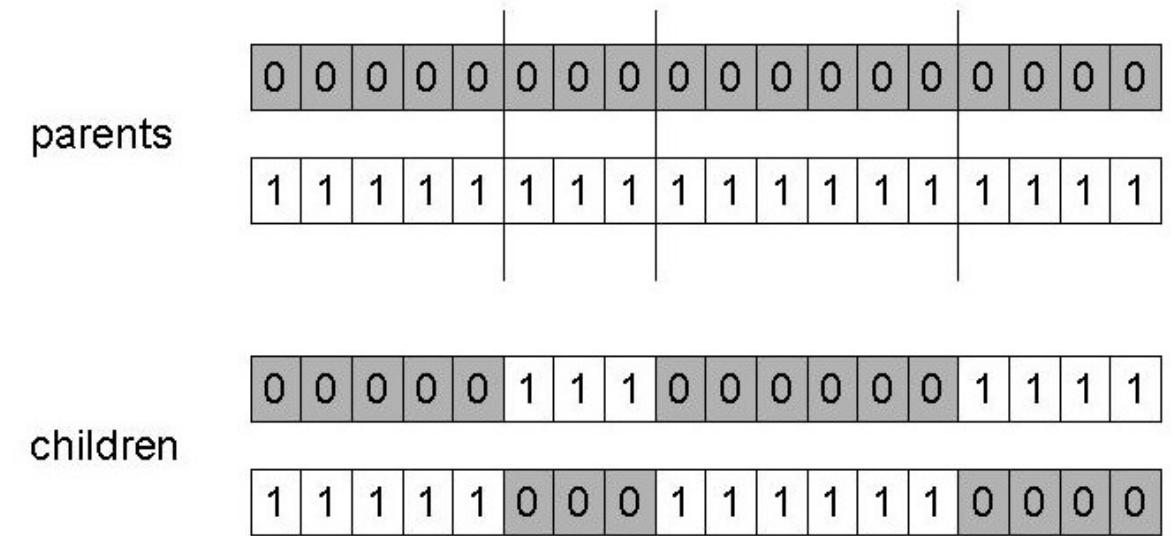
- Choose a random point on the **two** parents
- Split parents at this crossover point
- Create **two** children by exchanging tails



SGA - CROSSOVER

n-point crossover (the binary case)

- Choose n random crossover points,
- Split along those points,
- Glue parts, alternating between parents,
- Generalisation of 1 point (still some positional bias).



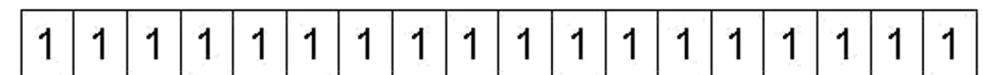
SGA - CROSSOVER

Uniform crossover (the binary case)

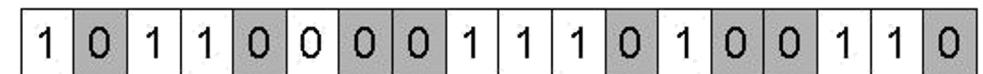
- Treats genes independently
 - Assign 'heads' to one parent, 'tails' to the other
 - Flip a coin for each gene of the first child
 - Make an inverse copy of the gene for the second child
 - Inheritance is independent of position.
 - Distributional bias:

avoids transmitting co-adapted genes

parents



children



Simple Genetic Algorithms

Mutation

SGA - MUTATION

Alter each gene independently with a probability p_m

p_m is called the mutation rate

- Typically between $1/\text{pop_size}$ and $1/\text{chromosome_length}$

parent

child

0	1	0	0	1	0	1	1	0	0	0	1	0	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

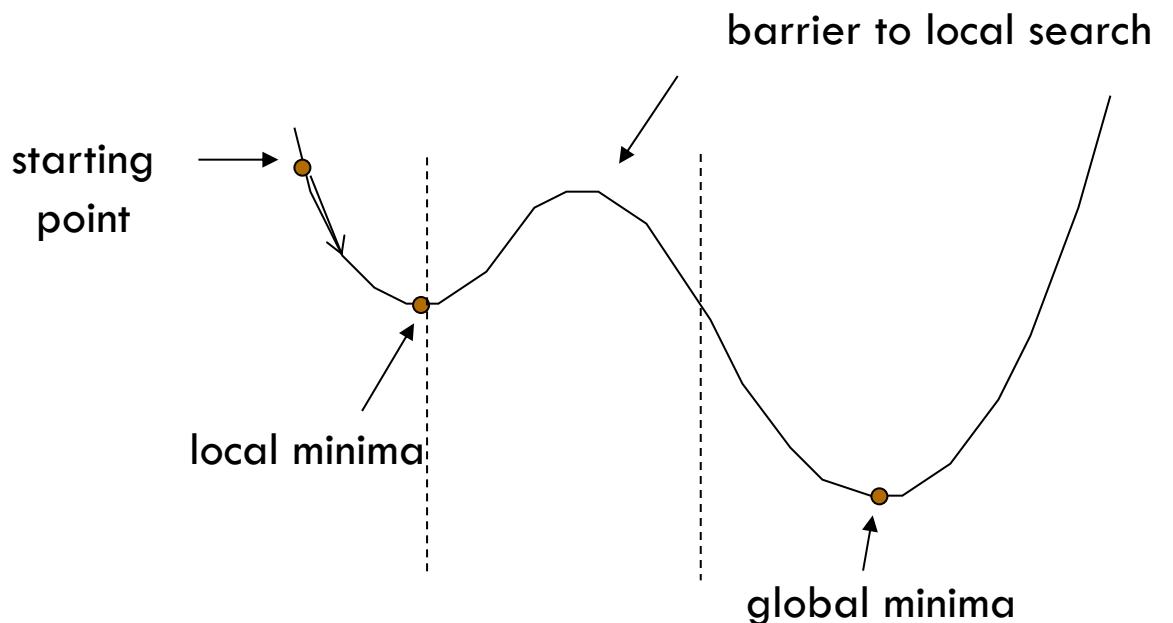
WHY CROSSOVER AND MUTATION?

Exploration:

Discovering promising areas in the search space.

Exploitation:

Optimizing within a promising area.



WHY CROSSOVER AND MUTATION?

Crossover is ***explorative***, it makes a big jump to an area somewhere between the two parent areas,

Mutation is ***exploitative***, it creates random small diversions, thereby staying near (in the area of) the parent.

Only crossover can combine information from two parents,

Only mutation can introduce new information.

Genetic Algorithms

Alternative Approaches

ALTERNATIVE BINARY REPRESENTATION

Gray coding of integers

- Gray coding is a mapping that means that small changes in the genotype cause small changes in the phenotype (unlike binary coding).
- “Smoothen” genotype-phenotype mapping makes life easier for the GA

Gray Code:

0	000
1	001
2	011
3	010
4	110
5	111
6	101
7	100

Hamming distance between any consecutive numbers is 1

GAS – OTHER REPRESENTATIONS

Other representations are also possible:

- Integers,
- Real-valued or floating-point numbers,
- Permutations.

The crossover and mutation actual implementation depends on the representation.

POPULATION MODELS

SGA uses a Generational model (GGA):

- Each individual survives for exactly one generation
- The entire set of parents is replaced by the offspring

Steady-State model:

- Only part of the population is replaced by the offspring,
- Usually one member of population is replaced,
- The proportion of the population replaced is called the ***Generational Gap***
 - 1.0 for GGA
 - $1/\text{pop_size}$ for SSGA

PARENTS AND OFFSPRING SELECTION

Selection can occur in two places:

- Selection from current generation to take part in mating (parent selection)
- Selection from parents + offspring to go into next generation (survivor selection)





REAL-VALUED GENETIC ALGORITHMS

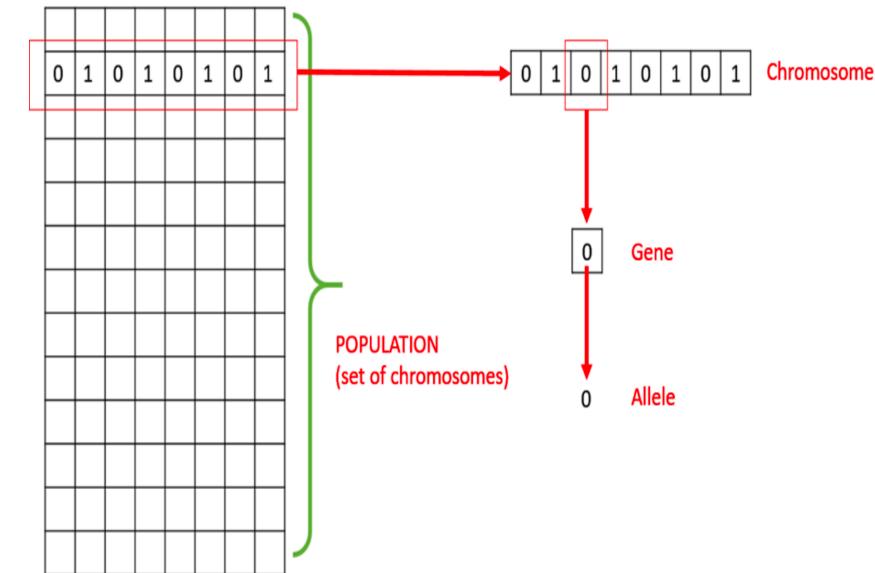
|| GA: STRATEGY AND MECHANICS..... SO FAR

Strategy: explore and exploit existing diversity in a population of solutions to achieve progressive solution enhancement.

Mechanics:

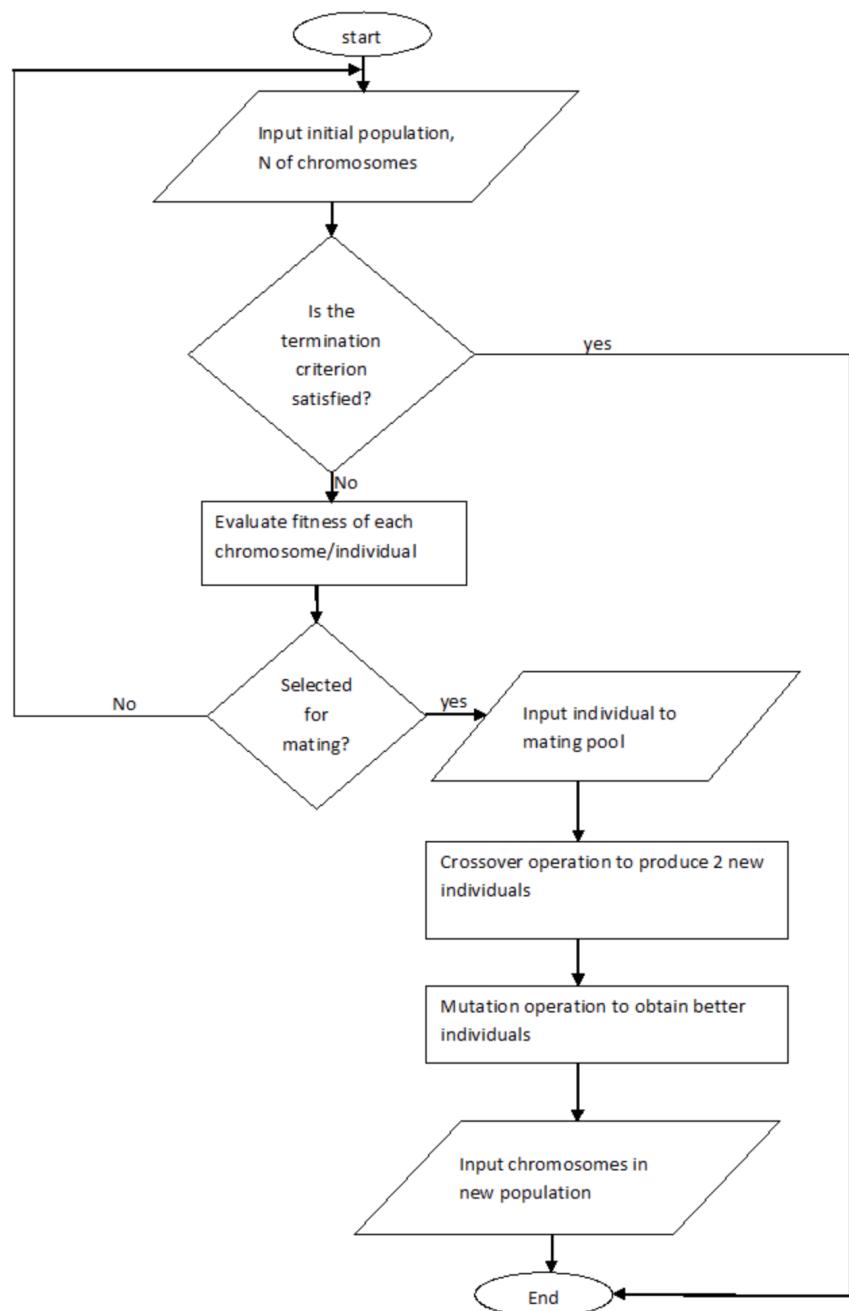
- Multi-candidate Solution creation or initiation- initial population.
 - Selection of off-spring next generation producers- parent selection
 - Selection criterion: Fitness
 - Next generation production: crossover
 - Next generation (parents) alteration/enhancement: mutation
 - Crossover **is explorative** and mutation **is exploitative**.
-
- Crossover:
 - 1-point crossover: Choose a random point on the two parents, split parents at this crossover point, exchanging tails
 - N-point crossover: Choose n random crossover points, split along those points, alternating between parents,
 - Uniform crossover: Assign 'heads' to one parent, 'tails' to the other, Flip a coin for each gene
 - Mutation: A GA operator. It alters each gene independently with a probability
 - Generation Gap: The proportion of the population replaced in each iteration.

Solution representation (chromosome): bits (genes) of binary values 0/1.



MECHANICS PROCESS

- We start with an initial population (which may be generated at random or seeded by other heuristics),
- Select parents from this population for mating.
- Apply crossover and mutation operators to generate new off-springs.
- Finally these off-springs replace (some or all) of the existing individuals in the population and the process repeats.
- In this way genetic algorithms actually try to mimic the human evolution to some extent.



REAL-VALUED GA'S

REAL-VALUED GAS

Many problems have real-valued parameters,

If mapped to a binary space, high precision solutions would require very long chromosomes,

How to combine and mutate real-valued solutions?

- Real-valued crossover
- Real-valued mutation

0	0	1	0	1	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---

Binary Representation



0.5	0.2	0.6	0.8	0.7	0.4	0.3	0.2	0.1	0.9
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Real-valued Representation

1	2	3	4	3	2	4	1	2	1
---	---	---	---	---	---	---	---	---	---

Integer Representation

1	5	9	8	7	4	2	3	6	0
---	---	---	---	---	---	---	---	---	---

Permutation Representation

REAL-VALUED GAS – CROSSOVER

Two types of real-valued crossover operators:

- **Discrete**: use any of the crossover operators identified before for binary representations
- **Intermediate (arithmetic)**:
 - Single arithmetic
 - Simple arithmetic
 - Whole arithmetic.

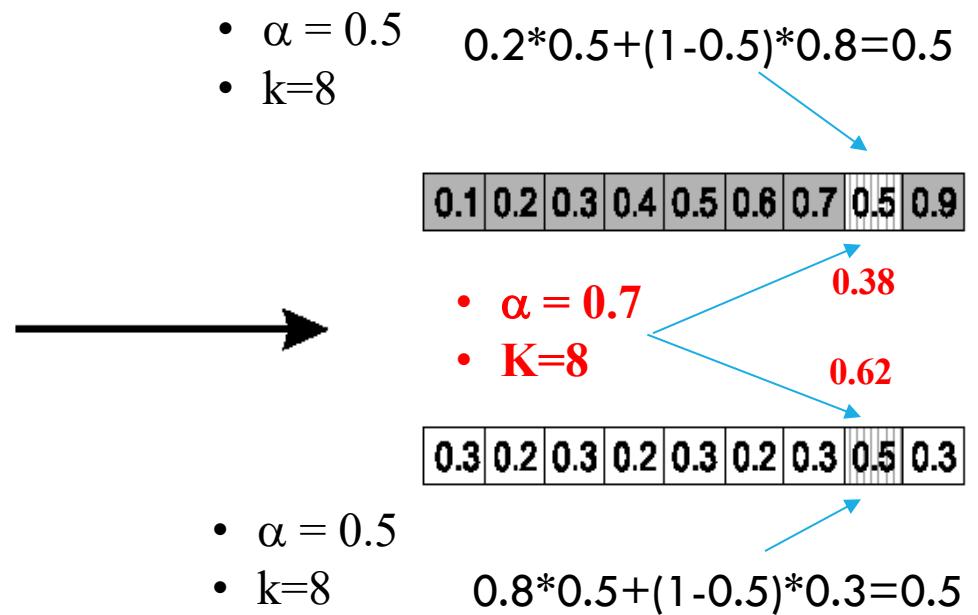
REAL-VALUED GAS – CROSSOVER

Single arithmetic crossover:

- Parents: $\langle x_1, \dots, x_n \rangle$ and $\langle y_1, \dots, y_n \rangle$
- Pick random gene (k) at random,
- Child 1 is: $\langle x_1, \dots, x_{k-1}, \alpha \cdot y_k + (1-\alpha) \cdot x_k, \dots, x_n \rangle$
- Reverse for other child.

Example:

0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
-----	-----	-----	-----	-----	-----	-----	-----	-----



REAL-VALUED GAS – CROSSOVER

Simple arithmetic crossover:

- Parents: $\langle x_1, \dots, x_n \rangle$ and $\langle y_1, \dots, y_n \rangle$,
- Pick random gene (k), after this point mix the values,
- Child 1 is:

$$\langle x_1, \dots, x_k, \alpha \cdot y_{k+1} + (1-\alpha) \cdot x_{k+1}, \dots, \alpha \cdot y_n + (1-\alpha) \cdot x_n \rangle$$

- Reverse for other child.

Example:

- $\alpha = 0.5$
- $k=6$
- $\alpha = 0.7$

0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
-----	-----	-----	-----	-----	-----	-----	-----	-----



0.42	0.38	0.48						
0.1	0.2	0.3	0.4	0.5	0.6	0.5	0.5	0.6
0.58	0.62	0.72						
0.3	0.2	0.3	0.2	0.3	0.2	0.5	0.5	0.6

REAL-VALUED GAS – CROSSOVER

Whole arithmetic crossover:

- Most commonly used
- Parents: $\langle x_1, \dots, x_n \rangle$ and $\langle y_1, \dots, y_n \rangle$
- Child 1 is: $a \cdot \bar{x} + (1 - a) \cdot \bar{y}$
- Reverse for other child.

Example:

0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
-----	-----	-----	-----	-----	-----	-----	-----	-----

0.2	0.2	0.3	0.3	0.4	0.4	0.5	0.5	0.6
-----	-----	-----	-----	-----	-----	-----	-----	-----



0.3	0.2	0.3	0.2	0.3	0.2	0.3	0.2	0.3
-----	-----	-----	-----	-----	-----	-----	-----	-----

0.2	0.2	0.3	0.3	0.4	0.4	0.5	0.5	0.6
-----	-----	-----	-----	-----	-----	-----	-----	-----

- $\alpha = 0.5$

REAL-VALUED GAS – MUTATION

The general scheme for mutation (called uniform mutation) is :

$$\bar{x} = \langle x_1, \dots, x_l \rangle \rightarrow \bar{x}' = \langle x'_1, \dots, x'_l \rangle$$
$$x_i, x'_i \in [LB_i, UB_i]$$

x'_i drawn randomly (uniform) from $[LB_i, UB_i]$

Analogous to bit-flipping mutation.

REAL-VALUED GAS – MUTATION

Another mutation scheme is to add random noise, for each gene:

$$x'_i = x_i + N(0, \sigma)$$

where $N(0, \sigma)$ is a random Gaussian number with mean zero and standard deviation σ .

This scheme falls into the non-uniform mutation schemes



PERMUTATIONS GENETIC ALGORITHMS

PERMUTATIONS GAS

For problems that take the form of deciding on the order in which a sequence of events should occur

Two types of problems exist:

- Events use limited resources or time. The **order** of events is important. e.g. Job Shop Scheduling,
- The **adjacency** of elements is important. e.g. Travelling Salesman Problem.

If there are n variables (e.g. cities) then the representation is a list of n integers, each of which occurs **exactly once**

5 4 3 2 1 6 10 9 8 7

PERMUTATIONS GAS – TSP EXAMPLE

Problem:

- Given n cities,
- Find a complete tour with minimal length.

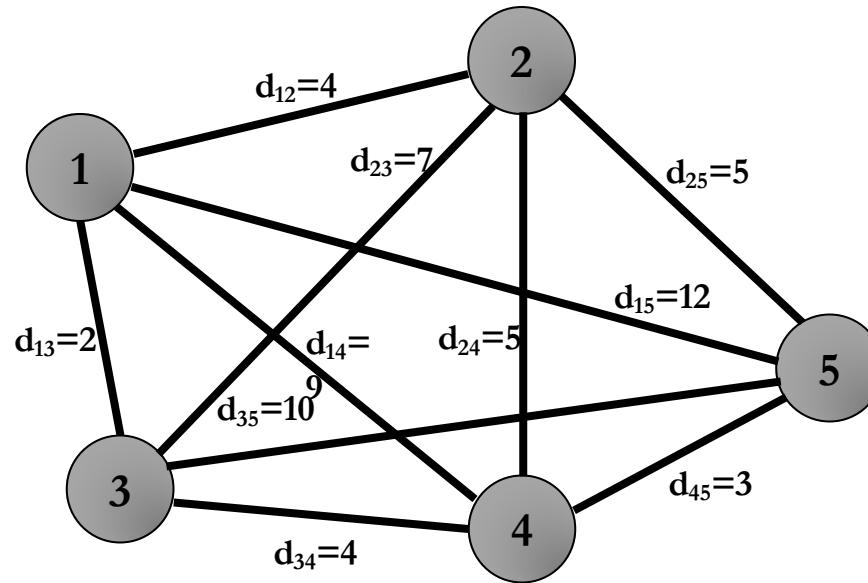
Encoding:

- Label the cities 1,2,...,n
- One complete tour is one permutation
 - [1,2,3,4], [3,4,2,1], ...

Search space is **BIG**: for n cities there are $(n-1)!$ possible tours.

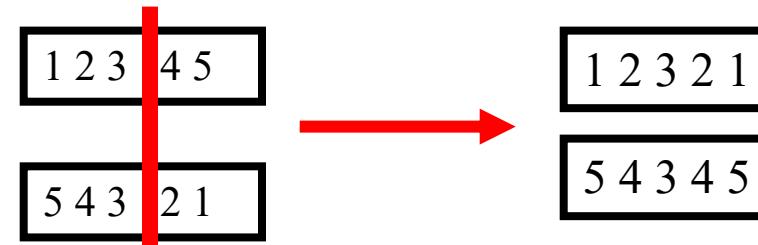


PERMUTATIONS GAS – TSP EXAMPLE



PERMUTATIONS GAS – TSP EXAMPLE

Previously defined crossover operators will often lead to inadmissible solutions,



Specialized crossover operators have been devised.

PERMUTATIONS GAS – TSP EXAMPLE

Four types of crossover operators:

- Adjacency-based:
 - **Partially Mapped Crossover (PMX),**
 - Edge crossover.
- Order-based:
 - **Order 1 crossover,**
 - **Cycle crossover.**

PERMUTATIONS GAS – TSP EXAMPLE



PMX: Informal procedure for parents **P1** and **P2**:

1. Choose random segment in child and copy it from **P1**,
2. Starting from the first crossover point look for elements in that segment of **P2** that have not been copied,
3. For each of these i , look in the offspring to see what element j has been copied in its place from **P1**,
4. Place i into the position occupied j in **P2**, since we know that we will not be putting j there (as is already in offspring).
5. If the place occupied by j in **P2** has already been filled in the offspring k , put i in the position occupied by k in **P2**,
6. Having dealt with the elements from the crossover segment, the rest of the offspring can be filled from **P2**. Second child is created analogously.

PERMUTATIONS GAS – TSP EXAMPLE

P1

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

Step 1



P2

9	3	7	8	2	6	5	1	4
---	---	---	---	---	---	---	---	---

Step 2

P1

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

P2

9	3	7	8	2	6	5	1	4
---	---	---	---	---	---	---	---	---



Step 3

P1

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

P2

9	3	7	8	2	6	5	1	4
---	---	---	---	---	---	---	---	---



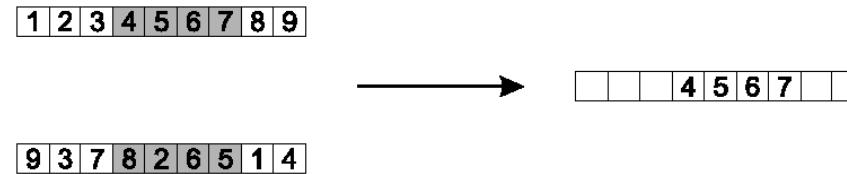
PERMUTATIONS GAS – TSP EXAMPLE

Order 1 crossover:

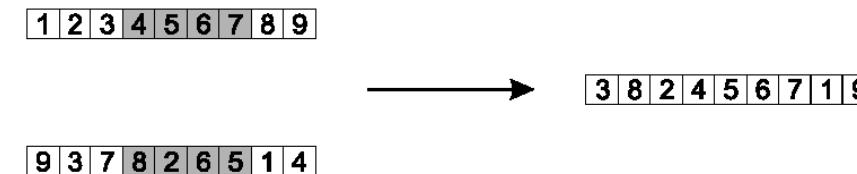
- The idea is to preserve relative order that elements occur
- Informal procedure:
 1. Choose an arbitrary part from the first parent,
 2. Copy this part to the first child,
 3. Copy the numbers that are not in the first part, to the first child:
 - Start right from cut point of the copied part,
 - Use the **order** of the second parent,
 - Wrap around at the end.
 4. Analogous for the second child, with parent roles reversed.

PERMUTATIONS GAS – TSP EXAMPLE

Copy randomly selected set from first parent



Copy rest from second parent in order 1,9,3,8,2



PERMUTATION GAS

Cycle Crossover :

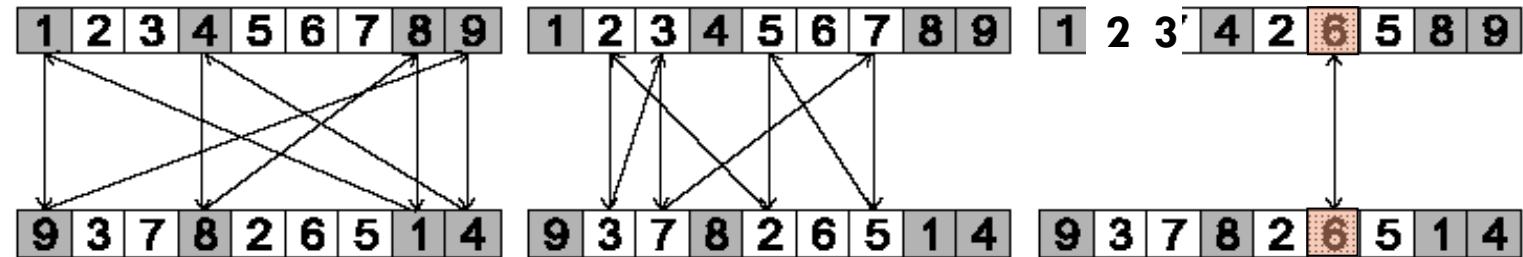
Each allele comes from one parent together with its position.

Informal procedure:

1. Make a cycle of alleles from P1 in the following way.
 - (a) Start with the first allele of P1.
 - (b) Look at the allele at the same position in P2.
 - (c) Go to the position with the same allele in P1.
 - (d) Add this allele to the cycle.
 - (e) Repeat step b through d until you arrive at the first allele of P1.
2. Put the alleles of the cycle in the first child on the positions they have in the first parent.
3. Take next cycle from second parent

CYCLE CROSSOVER EXAMPLE

Step 1: identify cycles



Step 2: copy alternate cycles into offspring

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

1	3	7	4	2	6	5	8	9
---	---	---	---	---	---	---	---	---



9	3	7	8	2	6	5	1	4
---	---	---	---	---	---	---	---	---

9	2	3	8	5	6	7	1	4
---	---	---	---	---	---	---	---	---

PERMUTATIONS GAS – TSP EXAMPLE

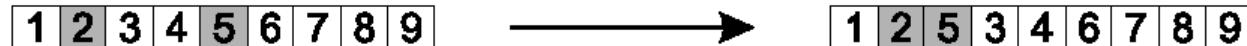
Four types of mutation operators:

- Insert mutation,
- Swap mutation,
- Inversion mutation,
- Scramble mutation.

PERMUTATIONS GAS – TSP EXAMPLE

Insert mutation:

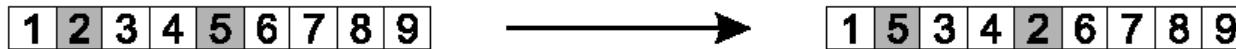
- Pick two genes values at random,
- Move the second to follow the first, shifting the rest along to accommodate
- Preserves most of the order and the adjacency information,



PERMUTATIONS GAS – TSP EXAMPLE

Swap mutation:

- Pick two genes at random and swap their positions,
- Preserves most of adjacency information (4 links broken), disrupts order more.



PERMUTATIONS GAS – TSP EXAMPLE

Inversion mutation:

- Pick two genes at random and then invert the substring between them,
- Preserves most adjacency information (only breaks two links) but disruptive of order information.



PERMUTATIONS GAS – TSP EXAMPLE

Scramble mutation:

- Pick two gene values at random,
- Randomly rearrange the genes in those positions (note subset does not have to be contiguous)

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---



1	3	5	4	2	6	7	8	9
---	---	---	---	---	---	---	---	---

PERMUTATIONS GAS – TSP EXAMPLE

The GA was applied for the berlin52 TSP instance (52 locations in Berlin) by having:

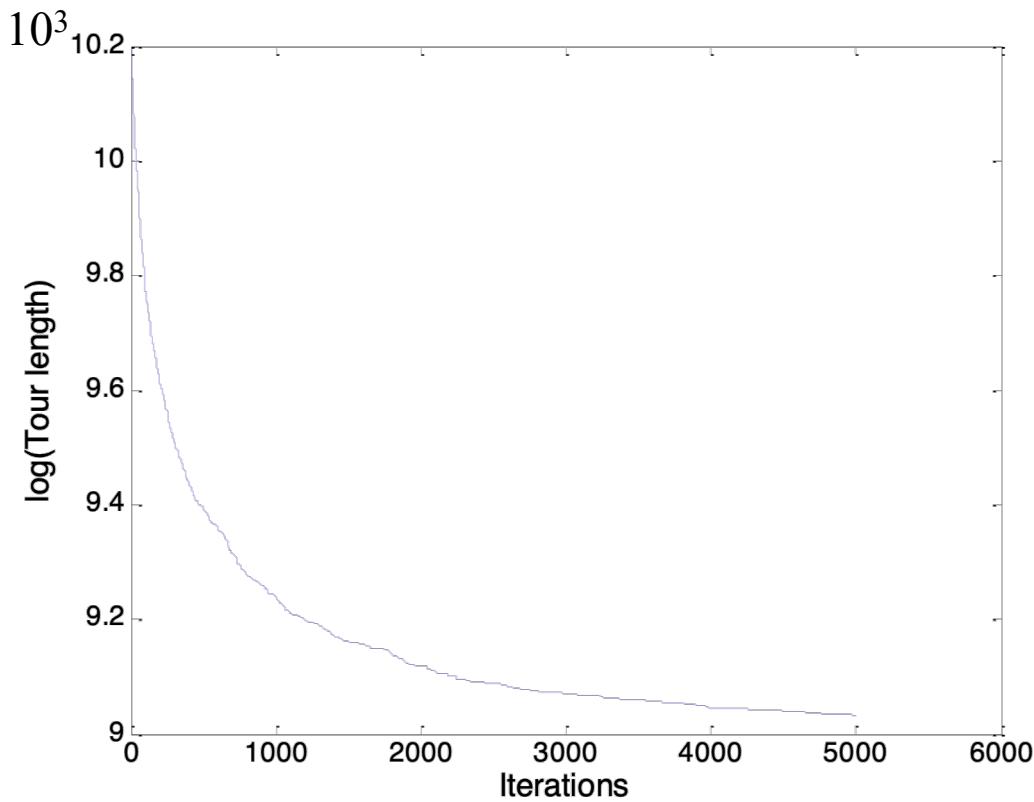
- 20 individuals
- Using order 1 crossover producing only one child
- Crossover probability = 0.7
- Using swap mutation with one swap only
- Mutation probability = 0.05
- Using the *elitism* model while keeping only the best individual from one population to the next,
- Using different number of iterations (500, 1000, 2000 and 5000),

PERMUTATIONS GAS – TSP EXAMPLE

Number of Iterations	Tour length
500	12135
1000	10069
2000	8986.2
5000	8730.3

SA obtained a tour of **8861**

PERMUTATIONS GAS – TSP EXAMPLE



A genetic algorithm-based approach to optimize the coverage and the localization in the wireless audio-sensors networks

Sami MNASRI

University of Toulouse
UT2J, CNRS-IRIT-IRT

Toulouse, France

Sami.Mnasri@fsgf.rnu.tn

Adel THALJAoui

University of Toulouse
UT2J, CNRS-IRIT-IRT

Toulouse, France

Adel.Thaljaoui@irit.fr

Nejah NASRI

University of Sfax
ENIS, ,LETI
Sfax, Tunisia

nejah.nasri@iseecs.rnu.tn

Thierry VAL

University of Toulouse
UT2J, CNRS-IRIT-IRT

Toulouse, France

val@irit.fr

Abstract—Coverage is one of the most important performance metrics for sensor networks that reflects how well a sensor field is monitored. In this paper, we are interested in studying the positioning and placement of sensor nodes in a WSN in order to maximize the coverage area and to optimize the audio localization in wireless sensor networks. First, we introduce the problem of deployment. Then we propose a mathematical formulation and a genetic based approach to solve this problem. Finally, we present the results of experimentations. This paper presents a genetic algorithm which aims at searching for an optimal or near optimal solution to the coverage holes problem. Compared with random deployment as well as existing methods, our genetic algorithm shows significant performance improvement in terms of quality.

We present the following model to resolve our problem. The objective is to provide a deployment scheme while optimizing the target coverage of the localization. To best locate, we aim to optimize the placement of nodes with the most possible uniform distribution of nodes (anchors and mobile nodes) around the target to locate. Among the considered constraints, the non-alignment of nodes and a well-studied distances between them. The set of targets to detect; the location of potential sites to install the sensors; the transmission power, the cost and the minimum number of received signals to detect a target are considered known in our model.

A. Assumptions

We set the following assumptions:

- Each anchor node is composed of two sensors (a bar containing two microphones), installed in such a manner that the bars of the different adjacent anchors are not aligned (Fig.1).
- Optimizing the localization considering that the target to be located must be within the audio range of at least two anchors nodes.
- There are two cases: either using two anchors, either using three anchors.
- When using two anchors, it is better to have a right angle between the two bars of microphones.
- When using three anchors, it is better that the mobile node is in the range of three anchors. Thus, each anchor must be oriented at 60 degrees with respect to each other.

B. Notation

The following notation is used in this paper. It is composed of sets, decision variables and parameters.

- **Sets**

- ✓ **T**: set of targets to detect in the field, tk is a target.
- ✓ **N**: the set of different types of sensor nodes, $N = \mathbf{Na} \cup \mathbf{Nb}$.
Na, the set of different types of stationary nodes
Nb the set of different types of mobile nodes
- ✓ **S**: set of potential sites to install the sensor nodes $S = \mathbf{Sa} \cup \mathbf{Sb}$
Sa the set of potential sites to install the stationary sensor nodes, na is a site of a stationary node.
Sb the set of potential sites to install the mobile sensor nodes, nm is a site of a mobile node.
(a site may not be in both sets, that is, $Sa \cap Sb \neq \emptyset$)

Decision Variables

- ✓ W_s^n be a 0-1 variable such that $W_s^n = 1$ if and only if a node of type $n \in N$ is installed at site $s \in S$
- ✓ X_{ts} , a 0-1 variable such that $X_{ts} = 1$ if the node of type $n \in N$ installed at site $s \in S$ receives a signal from a target at the position $t \in T$ with a power greater than or equal to the minimum required power by the node to detect it.
- ✓ $Sg_{ss'}$ is also a 0-1 variable such that $Sg_{ss'} = 1$ if and only if the node installed at site $s \in S$ receives a signal from another node installed at site $s' \in S$ with a power greater than or equal to the minimum required.
- ✓ $Sg_{ss'}^{nn'}$, be a 0-1 variable such that $Sg_{ss'}^{nn'} = 1$ if and only if the node of type $n \in N$ installed at site $s \in S$ receives a signal from another node of type $n' \in N$ installed at site $s' \in S$ with a power greater than or equal to the minimum required power.
- ✓ M_s is the minimum number of hops between a stationary node installed at site $s \in S$ to any mobile node.

Parameters

- ✓ γ_{ts} be the signal attenuation ratio from the target $t \in T$ to site $s \in S$,
- ✓ $\delta_{ss'}$ the attenuation ratio between the sites $s \in S$ and $s' \in S$,
- ✓ P_t is the transmission power of a target at the position $t \in T$ (in watts).
- ✓ p_n is the transmission power of a node of type $n \in N$ (in watts).
- ✓ P_{\min}^n is the minimum power of a received signal by a node of type $n \in N$ to detect it (i.e. the sensibility).
- ✓ n_{\min} the minimum number of nodes receiving a signal from a target to localize it (in our case, $n_{\min} \in \{2,3\}$),
- ✓ h_{\max} , the maximum number of hops between a anchor node and a mobile node,
- ✓ c_s the cost of a node of type $n \in N$ and installing it at site $s \in S$.
- ✓ Ag_{ij}^{m1m2} angle between two microphones $m1$ and $m2$ of two different and adjacent nodes i and j .
- ✓ n : length of the ROI (Region of interest).
- ✓ m : width of the ROI
- ✓ r : radius of a sensor (all the sensor nodes have the same sensing range).
- ✓ $nbNa$: number of stationary nodes
- ✓ $nbNm$: number of mobile nodes needed to add.
- ✓ nbT : number of targets
- ✓ Sg_{ij} : power of the signal transmitted between two nodes i and j .
- ✓ d_{ij} : distance between two nodes i and j ,
- ✓ d_{m1m2} : a constant representing the distance between two sensors (microphones) of the same node.
- ✓ d_{\max} : a constant representing the maximum distance between two nodes i and j (or a node i and a target j) so that they can be detectable.

C. Objective function

To model the problem of target coverage considering the localization, we consider the following objective function.

- Coverage: Let F_1 be the fitness of a mobile node i (n_{mi}) which calculates the coverage as a function of the targets it covered, we obtain the following function F_1 for the coverage

$$F_1 = \text{Maximize} \left(\sum_{n_m \in Nm} F(n_m) \right) \quad (1)$$

- Localization: each target must be monitored by at least n_{min} nodes (mobile or anchor), thus:

$\sum_{s \in S} x_{ts} \geq n_{min} \forall t \in T$, we obtain the following function F_2 for the localization:

$$F_2 = \text{Maximize} \left(\sum_{t \in T} \left(\sum_{s \in S} x_{ts} - n_{min} \right)^+ \right)$$

knowing that $(x)^+ = \max(0, x)$ (2)

Thus, the fitness function is given by:

$$F = F_1 + F_2$$

$$= \text{Maximize} \sum_{t \in T} \left(\sum_{s \in S} x_{ts} - n_{min} \right)^+ + \sum_{n \in N} F(n_m) \quad (3)$$

D. Constraints

F is subject to:

$$\sum_{s \in S} x_{ts} \geq n_{min} \forall t \in T$$

$$Sg_{ts} = 1 \Rightarrow Ag_{ts}^{m1m2} = k\Pi / n_{min}, t \neq s$$

$$Sg_{ts} = 1 \Rightarrow \sum_{n \in N} Ag_{ts}^{m1m2} \in [0. k\Pi], t \neq s \quad (6)$$

$$nbNa = n_{min} \cdot (nm / 2\Pi r^2) \quad (7)$$

$$d_{ts} = \alpha \cdot Sg_{ts} \cdot g(Sg_{ts}), \alpha \in R \quad (8)$$

$$(Sg_{ts} = 1) \Rightarrow (d_{ts} \leq d_{max}) \quad (9)$$

$$\sum_{s \in S} x_{ts} \leq \sum_{n \in N} W_s^n \quad (10)$$

$$\delta_{ss'} \sum_{n \in N} P^n \sum_{n' \in N} Sg_{ss'}^{nn'} \geq \sum_{n' \in N} P_{\min}^{n'} \sum_{n \in N} Sg_{ss'}^{nn'} \quad (11)$$

$$Sg_{ss'} = \sum_{n \in N} \sum_{n' \in N} Sg_{ss'}^{nn'} \quad (12)$$

$$\delta_{ts} P_t \geq \sum_{n \in N} P_{\min}^n W_s^n, \forall t \in T, s \in S \quad (13)$$

The objective function (3) of the problem aims to optimize the target coverage and the localization. **Constraint (4)** impose that the number of nodes receiving a signal from the target i must be greater than or equal to the minimum necessary to localize it. **Constraint (5)** force the angles of arrival between sensors (microphones) to be 90° in the case of 2-coverage ($n_{\min}=2$) and to be 60° in the case of 3-coverage ($n_{\min}=3$). **Constraint (6)** concerns the non-linearity of the adjacent nodes in order to optimize the localization. **Constraint (7)** imposes the number of the anchors deployed initially. **Constraint (8)** link the distance and the power transmission of the signal between two nodes. g is a function, α is real coefficient. **Constraint (9)**, imply that if there is a signal S_{gt} between two nodes, the distance between these two nodes (d_{ij}) should not exceed a fixed maximum distance (d_{\max}). **Constraint (10)**, impose that a target cannot be detected by a number of nodes that exceeds the number of installed nodes in the different sites. **Constraint (11)** impose that if the node s' is detected by the node s , then the power transmission resulting from s towards s' must be higher than the minimum necessary power transmission so that s is detectable by s' . **Constraint (12)** concerns the power transmission emitted by the node s and received by the node s' , for different types of nodes. **Constraint (13)** indicate that the node installed at site s must receive a signal from a target at position t with a power greater than or equal to the minimum required to detect it.

B. Coverage Model

We assume that each sensor node has a sensing radius r which covers a circular area. We also assume that a target t_k can be detected by the sensor S_i if t_k is within the sensing range of S_i . We also assumed that d is the distance between the target object being sensed t_k and the sensor node S_i . The coverage function $\text{Coverage}(S)$ is equal to 1 if the target object can be sensed and covered; otherwise it is equal to 0. This binary model of sensor detection can represented as follows:

$$\text{Coverage}(S) = \begin{cases} 1, & d(S_i, t_k) \leq r \\ 0, & d(S_i, t_k) > r \end{cases}$$

- Representing a chromosome

In the proposed genetic algorithm a chromosome represents a solution that indicates the position (location) of a potential mobile node in the region of interest (RoI). This position is modeled as an (X, Y) point. The different gens of the chromosome represent a binary digit that resembles the value of the position on the X and Y axes. For example, to represent a mobile node mapped to the location (50, 65), the corresponding chromosome is shown in Fig.2. The Choice of the size of the chromosome population is based on two factors: the area of the RoI and the initial configuration of the network. For instance, if the radius of each node is 48m and the area of the sensing field is 70 m * 80 m, the number of deployed stationary nodes will be (i.e.; $(70*80)/(\pi \cdot 48^2) \approx 117$), then the algorithm will start with population of 117 randomly generated chromosomes to ensure the full coverage. The value 117 is selected based on the assumption that 117 sensor nodes would cover the entire field as if they were deterministically deployed. If we aim to ensure a k-coverage (each target must be covered by at least k sensor nodes), we have to start with $117 * k$ chromosomes as an initial population.

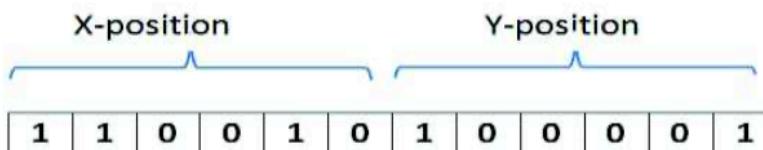


Fig. 2. Chromosome representing the sensor position (50, 65)

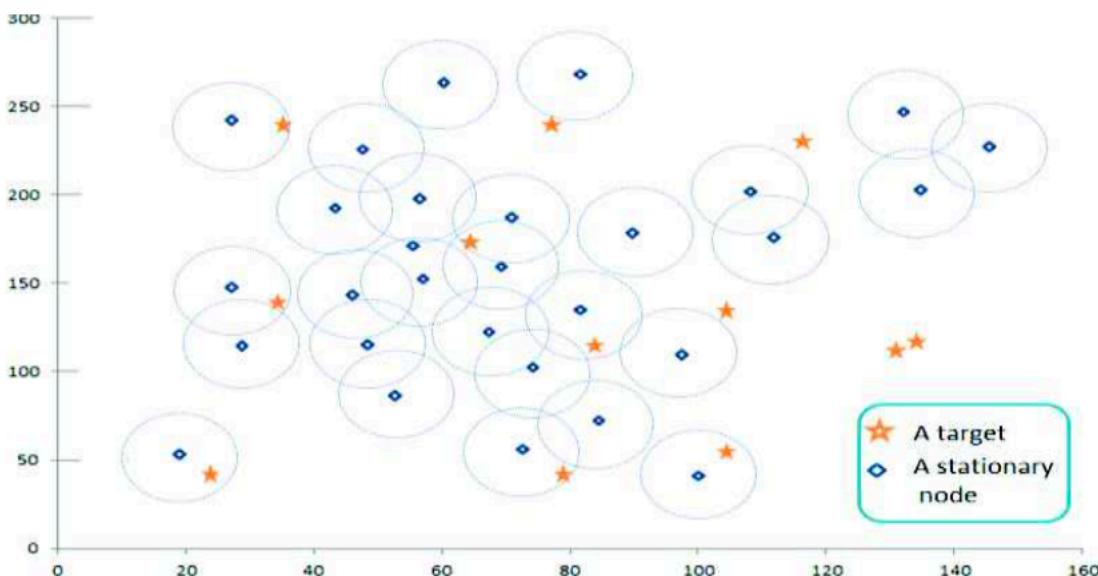


Fig. 3. Initial Random Deployment

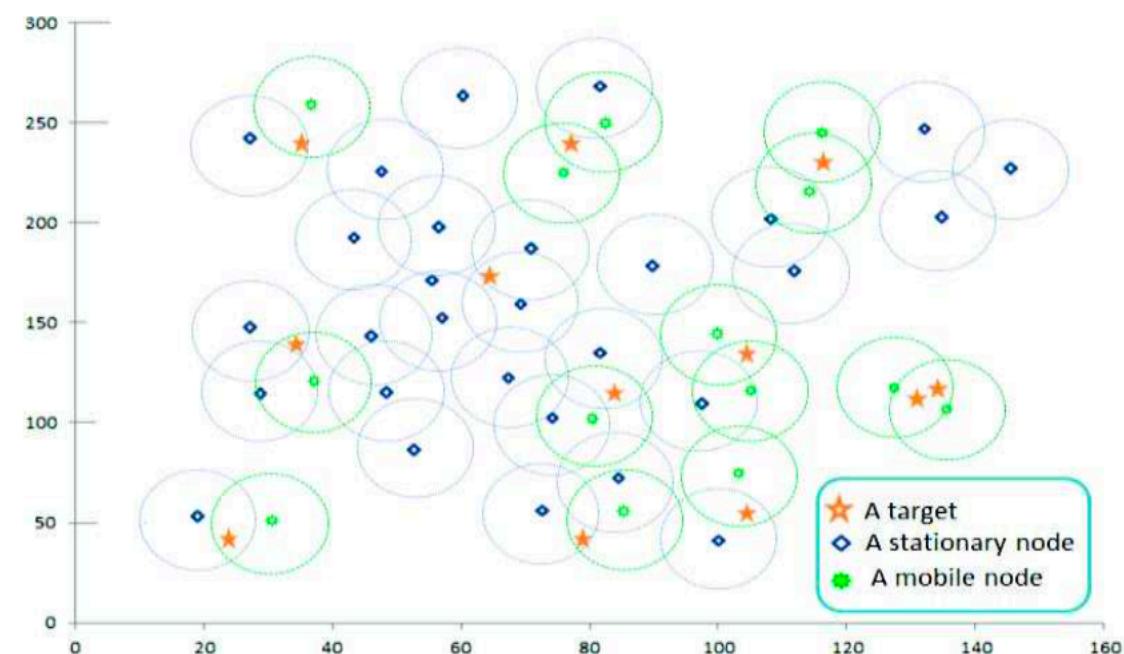


Fig. 4. Deployment after executing the GA for the 2-coverage case

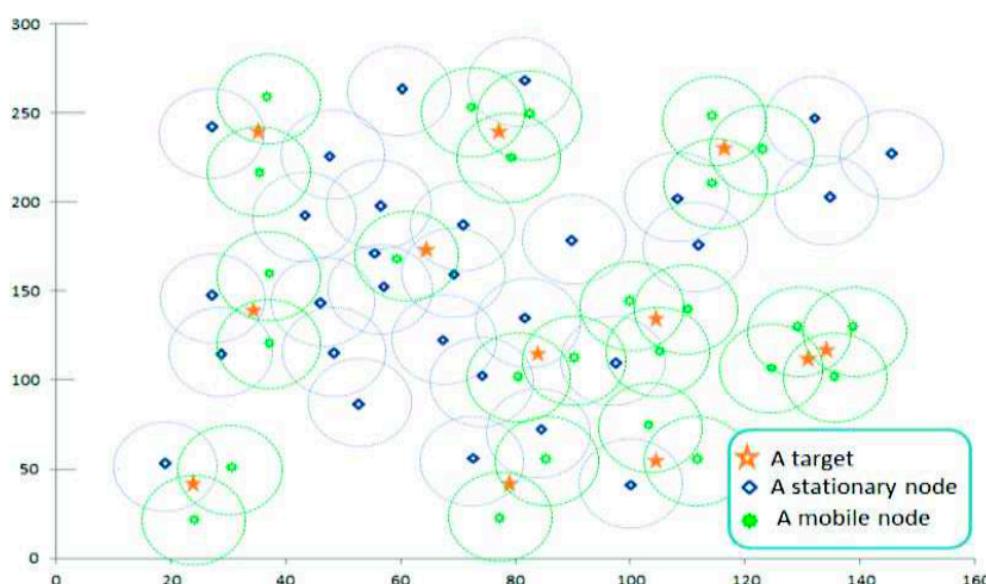


Fig. 5. Deployment after executing the GA for the 3 –coverage case



Open Archive TOULOUSE Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in : <http://oatao.univ-toulouse.fr/>
Eprints ID : 15237

The contribution was presented at ISNCC 2015:

<http://www.isncc-conf.org/>

To cite this version : Mnasri, Sami and Thaljaoui, Adel and Nasri, Nejah and Val, Thierry *A genetic algorithm-based approach to optimize the coverage and the localization in the wireless audiosensors networks.* (2015) In: IEEE International Symposium on Networks, Computers and Communications (ISNCC 2015), 13 May 2015 - 15 May 2015 (Hammamet, Tunisia).

Any correspondence concerning this service should be sent to the repository administrator: staff-oatao@listes-diff.inp-toulouse.fr

A genetic algorithm-based approach to optimize the coverage and the localization in the wireless audio-sensors networks

Sami MNASRI
University of Toulouse
UT2J, CNRS-IRIT-IRT
Toulouse, France
Sami.Mnasri@fsgf.rnu.tn

Adel THALJAoui
University of Toulouse
UT2J, CNRS-IRIT-IRT
Toulouse, France
Adel.Thaljaoui@irit.fr

Nejah NASRI
University of Sfax
ENIS, ,LETI
Sfax, Tunisia
nejah.nasri@iseecs.rnu.tn

Thierry VAL
University of Toulouse
UT2J, CNRS-IRIT-IRT
Toulouse, France
val@irit.fr

Abstract—Coverage is one of the most important performance metrics for sensor networks that reflects how well a sensor field is monitored. In this paper, we are interested in studying the positioning and placement of sensor nodes in a WSN in order to maximize the coverage area and to optimize the audio localization in wireless sensor networks. First, we introduce the problem of deployment. Then we propose a mathematical formulation and a genetic based approach to solve this problem. Finally, we present the results of experimentations. This paper presents a genetic algorithm which aims at searching for an optimal or near optimal solution to the coverage holes problem. Compared with random deployment as well as existing methods, our genetic algorithm shows significant performance improvement in terms of quality.

Keywords— Target Coverage, Audio Localization, Mobile Node, Deployment; Genetic Algorithm; NSGAII

I. INTRODUCTION

Coverage area can be defined, according to [1], as: "the area in which a sensor can perform its sensing, monitoring, surveillance and detection tasks with a reasonable accuracy (i.e., the sensor reading have at least a threshold level of sensing detection probabilities within the area). The target coverage (called also point coverage) interest in controlling a target in the field of interest that can be stationary or mobile. The k-coverage problem requires preserving at least k sensor nodes controlling any target to consider it covered. The works of [2] present and discuss the types of coverage problems. The localization of the sensors is the most significant factor related to the cover network. Also, localization is an important issue when there is an uncertainty of the exact position of some nodes. Indeed, in wireless sensor networks, the location information is crucial especially when an unusual event occurs. In this case, sensor node that detected that event needs to locate it and then report this position to the base station.

The use of acoustic information captured by sensor nodes is one of the axes that can bring more possibilities in term of

localization. In our work, time difference of arrival (TDoA) using correlation technique was used for estimating the delay between two signals captured by two different microphones placed on one node. The direction of arrival of the sound source can be obtained using this delay and the sound source is positioned by adopting the geometric location method.

For most deployment formulations, the problem of optimal placement of the sensor nodes is proven NP-hard [3]. Consequently, for large scale instances, this problem cannot be solved by deterministic methods such as the circle packing algorithm. We define the problem formally and we propose an efficient genetic algorithm to resolve the problem of the coverage holes after the initial random deployment. For a given number of sensors, the proposed algorithm attempts to maximize the sensor field coverage using a set of operators.

In our works, we are interesting in using WSN in smart buildings applications. Despite the different challenges in WSNs, research works have only focused on post-deployment problems such as: sensors localization, MAC efficiency or routing optimization, etc. Our works aim to ensure the deployment of the nodes while maximizing the coverage and optimizing the audio localization using an efficient genetic algorithm. Our proposed model is different from the existing models since it integrates sensor node deployment, and audio localization approach in a single model.

The rest of the paper is organized as follows: In Section II a mathematical modeling is proposed. In Section III, the genetic algorithm based approach is explained. In Section IV, the target localization issues are discussed. In Section V, numerical results are presented and discussed and finally, Section VI concludes this paper.

II. RELATED WORKS

Different research works deals about the deployment problem in order to maximize the coverage in WSN. The works of [4] and [5] interests in studying the sensor deployment problems. Also, in [6], the coverage problem is studied in the domain of the robot exploration. This work considers each robot as a sensor node and the used algorithm

deploy nodes one by one incrementally. Hence, the proposed algorithm is computationally expensive, when we increase the number of nodes. Some recent researches proposed genetic algorithms to resolve the deployment problem in WSNs. As example, the works of [7] propose a multi-objective paradigm to solve the deployment and power assignment problem. This evolutionary algorithm is based on the MOEA/D (Multi Objective Evolutionary Algorithm/Decomposition). They gave a comparison between the MOEA/D algorithm and the NSGAII algorithm. The former is better in some instances while the latter is better in some other instances.

III. MATHEMATICAL MODEL

We present the following model to resolve our problem. The objective is to provide a deployment scheme while optimizing the target coverage of the localization. To best locate, we aim to optimize the placement of nodes with the most possible uniform distribution of nodes (anchors and mobile nodes) around the target to locate. Among the considered constraints, the non-alignment of nodes and a well-studied distances between them. The set of targets to detect; the location of potential sites to install the sensors; the transmission power, the cost and the minimum number of received signals to detect a target are considered known in our model.

A. Assumptions

We set the following assumptions:

- Each anchor node is composed of two sensors (a bar containing two microphones), installed in such a manner that the bars of the different adjacent anchors are not aligned (Fig.1).
- Optimizing the localization considering that the target to be located must be within the audio range of at least two anchors nodes.
- There are two cases: either using two anchors, either using three anchors.
- When using two anchors, it is better to have a right angle between the two bars of microphones.
- When using three anchors, it is better that the mobile node is in the range of three anchors. Thus, each anchor must be oriented at 60 degrees with respect to each other.

B. Notation

The following notation is used in this paper. It is composed of sets, decision variables and parameters.

• Sets

- ✓ T : set of targets to detect in the field, t_k is a target.
- ✓ N : the set of different types of sensor nodes, $N = \mathbf{Na} \cup \mathbf{Nb}$.
 \mathbf{Na} , the set of different types of stationary nodes
 \mathbf{Nb} the set of different types of mobile nodes
- ✓ S : set of potential sites to install the sensor nodes $S = \mathbf{Sa} \cup \mathbf{Sb}$
 \mathbf{Sa} the set of potential sites to install the stationary sensor nodes, na is a site of a stationary node.

\mathbf{Sb} the set of potential sites to install the mobile sensor nodes, nm is a site of a mobile node.
(a site may not be in both sets, that is, $\mathbf{Sa} \cap \mathbf{Sb} \neq \emptyset$)

• Decision Variables

- ✓ W_s^n be a 0-1 variable such that $W_s^n = 1$ if and only if a node of type $n \in N$ is installed at site $s \in S$
- ✓ X_{ts} , a 0-1 variable such that $X_{ts} = 1$ if the node of type $n \in N$ installed at site $s \in S$ receives a signal from a target at the position $t \in T$ with a power greater than or equal to the minimum required power by the node to detect it.
- ✓ $Sg_{ss'}$ is also a 0-1 variable such that $Sg_{ss'} = 1$ if and only if the node installed at site $s \in S$ receives a signal from another node installed at site $s' \in S$ with a power greater than or equal to the minimum required.
- ✓ $Sg_{ss'}^{nn'}$, be a 0-1 variable such that $Sg_{ss'}^{nn'} = 1$ if and only if the node of type $n \in N$ installed at site $s \in S$ receives a signal from another node of type $n' \in N$ installed at site $s' \in S$ with a power greater than or equal to the minimum required power.
- ✓ M_s is the minimum number of hops between a stationary node installed at site $s \in S$ to any mobile node.

• Parameters

- ✓ γ_{ts} be the signal attenuation ratio from the target $t \in T$ to site $s \in S$,
- ✓ $\delta_{ss'}$ the attenuation ratio between the sites $s \in S$ and $s' \in S$,
- ✓ P_t is the transmission power of a target at the position $t \in T$ (in watts).
- ✓ p_n is the transmission power of a node of type $n \in N$ (in watts).
- ✓ P_{\min}^n is the minimum power of a received signal by a node of type $n \in N$ to detect it (i.e. the sensibility).
- ✓ n_{\min} the minimum number of nodes receiving a signal from a target to localize it (in our case, $n_{\min} \in \{2,3\}$),
- ✓ hp_{\max} , the maximum number of hops between a anchor node and a mobile node,
- ✓ c_s the cost of a node of type $n \in N$ and installing it at site $s \in S$.
- ✓ Ag_{ij}^{m1m2} angle between two microphones $m1$ and $m2$ of two different and adjacent nodes i and j .

- ✓ n: length of the RoI (Region of interest).
- ✓ m: width of the RoI
- ✓ r: radius of a sensor (all the sensor nodes have the same sensing range).
- ✓ nbNa: number of stationary nodes
- ✓ nbNm: number of mobile nodes needed to add.
- ✓ nbT: number of targets
- ✓ Sg_{ij} : power of the signal transmitted between two nodes i and j.
- ✓ d_{ij} : distance between two nodes i and j,
- ✓ d_{m1m2} : a constant representing the distance between two sensors (microphones) of the same node.
- ✓ d_{max} : a constant representing the maximum distance between two nodes i and j (or a node i and a target j) so that they can be detectable.

C. Objective function

To model the problem of target coverage considering the localization, we consider the following objective function.

- Coverage: Let F1 be the fitness of a mobile node i (n_{mi}) which calculates the coverage as a function of the targets it covered, we obtain the following function F1 for the coverage

$$F1 = \text{Maximize} \left(\sum_{n_m \in N_m} F(n_m) \right) \quad (1)$$

- Localization: each target must be monitored by at least n_{min} nodes (mobile or anchor), thus:

$$\sum_{s \in S} x_{ts} \geq n_{min} \forall t \in T, \text{ we obtain the following function F2 for the localization:}$$

$$F2 = \text{Maximize} \left(\sum_{t \in T} \left(\sum_{s \in S} x_{ts} - n_{min} \right)^+ \right) \quad (2)$$

knowing that $(x)^+ = \max(0, x)$

Thus, the fitness function is given by:

$$\begin{aligned} F &= F1 + F2 \\ &= \text{Maximize} \\ &\quad \sum_{t \in T} \left(\sum_{s \in S} x_{ts} - n_{min} \right)^+ + \sum_{n \in N} F(n_m) \end{aligned} \quad (3)$$

D. Constraints

F is subject to:

$$\sum_{s \in S} x_{ts} \geq n_{min} \forall t \in T \quad (4)$$

$$Sg_{ts} = 1 \Rightarrow Ag_{ts}^{m1m2} = k\Pi / n_{min}, t \neq s \quad (5)$$

$$Sg_{ts} = 1 \Rightarrow \sum_{n \in N} Ag_{ts}^{m1m2} \in [0, k\Pi], t \neq s \quad (6)$$

$$nbNa = n_{min} \cdot (nm / 2\Pi r^2) \quad (7)$$

$$d_{ts} = \alpha \cdot Sg_{ts} \cdot g(Sg_{ts}), \alpha \in R \quad (8)$$

$$(Sg_{ts} = 1) \Rightarrow (d_{ts} \leq d_{max}) \quad (9)$$

$$\sum_{s \in S} x_{ts} \leq \sum_{n \in N} W_s^n \quad (10)$$

$$\delta_{ss'} \sum_{n \in N} P^n \sum_{n' \in N} Sg_{ss'}^{nn'} \geq \sum_{n' \in N} P_{\min}^{n'} \sum_{n \in N} Sg_{ss'}^{nn'} \quad (11)$$

$$Sg_{ss'} = \sum_{n \in N} \sum_{n' \in N} Sg_{ss'}^{nn'} \quad (12)$$

$$\delta_{ts} P_t \geq \sum_{n \in N} P_{\min}^n W_s^n, \forall t \in T, s \in S \quad (13)$$

The objective function (3) of the problem aims to optimize the target coverage and the localization. **Constraint (4)** impose that the number of nodes receiving a signal from the target i must be greater than or equal to the minimum necessary to localize it. **Constraint (5)** force the angles of arrival between sensors (microphones) to be 90° in the case of 2-coverage ($n_{min}=2$) and to be 60° in the case of 3-coverage ($n_{min}=3$). **Constraint (6)** concerns the non-linearity of the adjacent nodes in order to optimize the localization. **Constraint (7)** imposes the number of the anchors deployed initially. **Constraint (8)** link the distance and the power transmission of the signal between two nodes. g is a function, α is real coefficient. **Constraint (9)**, imply that if there is a signal Sg_{ts} between two nodes, the distance between these two nodes (d_{ij}) should not exceed a fixed maximum distance (d_{max}). **Constraint (10)**, impose that a target cannot be detected by a number of nodes that exceeds the number of installed nodes in the different sites. **Constraint (11)** impose that if the node s' is detected by the node s , then the power transmission resulting from s towards s' must be higher than the minimum necessary power transmission so that s is detectable by s' . **Constraint (12)** concerns the power transmission emitted by the node s and received by the node s' , for different types of nodes. **Constraint (13)** indicate that the node installed at site s must receive a signal from a target at position t with a power greater than or equal to the minimum required to detect it.

IV. TARGET LOCALIZATION

Let's consider a mobile source emitting a sound $s(t)$ and a node equipped with two microphones. Each one of the two microphones is receiving a signal ($s_1(t)$ for the first and $s_2(t)$ for the second). Due the distance between the two microphones, a difference of time between the observations of the sound signal will be noted at each microphone, referred to as Time difference of Arrival. TDoA (Fig.1) is computed using the spatial positions of the target and microphones.

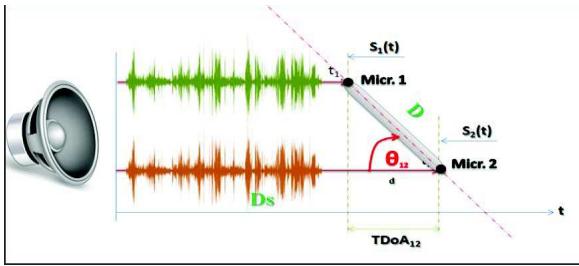


Fig. 1. Time Difference of Arrival (TDoA)

Acoustic localization is done following two major steps; The first step consists on estimating the time difference of arrival (TDoA) of the signals captured by two separated microphones of one node. Then, the direction of arrival of the sound with respect to this node is computed using trigonometry specifications. The second step consists on localizing the acoustic source using at least two nodes. The process consists on merging the results obtained with each node in term of direction of arrival and then use a specific geometric positioning method in order to compute the geometric coordinates of the acoustic source in a 2D space

V. A GENETIC ALGORITHM FOR THE DEPLOYMENT CONSIDERING THE TARGET LOCALIZATION

In this section, we present the suggested approach. We present the assumptions of the network, the coverage model, and we discuss the approach based on the genetic algorithm.

A. Network Assumptions

We assume that the sensor nodes are randomly deployed and the number of sensor nodes initially deployed is equal to the required number to achieve nmin-coverage ($n_{\min} \in \{2,3\}$) as if these nodes were deterministically deployed. We also assumed that mobile nodes are used to repair the coverage holes after the initial deployment of the stationary nodes.

B. Coverage Model

We assume that each sensor node has a sensing radius r which covers a circular area. We also assume that a target t_k can be detected by the sensor S_i if t_k is within the sensing range of S_i . We also assumed that d is the distance between the target object being sensed t_k and the sensor node S_i . The coverage function $\text{Coverage}(S)$ is equal to 1 if the target object can be sensed and covered; otherwise it is equal to 0. This binary model of sensor detection can represented as follows:

$$\text{Coverage}(S) = \begin{cases} 1, & d(S_i, t_k) \leq r \\ 0, & d(S_i, t_k) > r \end{cases}$$

C. The Proposed Genetic Algorithm (NSGAII)

We aim at maximizing the coverage rate by reducing the holes, and maximizing the localization.. Assuming that S_i is the stationary sensor nodes deployed randomly over the region of interest, r is the sensing range of the sensors. The proposed genetic algorithm starts with an initial random population (the distribution of the initial nodes). Then, the objective function evaluates in each iteration the constraints satisfaction rate. The new solution (population) is improved after each iteration of the algorithm. This improvement is carried out through the

operators (crossover and mutation). A stopping criterion is used to stop the execution of the algorithm. The genetic algorithm is run by the base station after gathering the positions of the stationary nodes in order to determine the number and positions of the mobile nodes as follow:

• Representing a chromosome

In the proposed genetic algorithm a chromosome represents a solution that indicates the position (location) of a potential mobile node in the region of interest (RoI). This position is modeled as an (X, Y) point. The different gens of the chromosome represent a binary digit that resembles the value of the position on the X and Y axes. For example, to represent a mobile node mapped to the location (50, 65), the corresponding chromosome is shown in Fig.2. The Choice of the size of the chromosome population is based on two factors: the area of the RoI and the initial configuration of the network. For instance, if the radius of each node is 48m and the area of the sensing field is $70 \text{ m} * 80 \text{ m}$, the number of deployed stationary nodes will be (i.e.; $(70*80)/(\pi \cdot 48^2) \approx 117$), then the algorithm will start with population of 117 randomly generated chromosomes to ensure the full coverage. The value 117 is selected based on the assumption that 117 sensor nodes would cover the entire field as if they were deterministically deployed. If we aim to ensure a k-coverage (each target must be covered by at least k sensor nodes), we have to start with $117 * k$ chromosomes as an initial population.

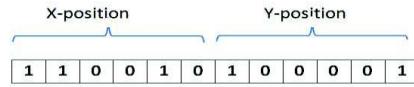


Fig. 2. Chromosome representing the sensor position (50, 65)

• Evaluation

After the initialization, each chromosome fitness (i.e.; the goodness of the solution) is evaluated using the fitness function. The fitness or the formulation of the objective depends on characteristics of the problem. The fitness function is used to choose the best fittest chromosomes to reproduce the next generated solutions by the algorithm. The fitness function calculates the maximum number of the covered targets by each mobile node. The overlapping redundancy is prevented by the fitness function among the coverage regions of the deployed mobile nodes. The fitness function is given by:

$$F = \text{Maximize} \left(\sum_{t \in T} \left(\sum_{s \in S} x_{ts} - n_{\min} \right)^+ + \sum_{n \in N} F(n_m) \right) \quad (14)$$

• Reproduction

Reproduction is composed of four steps: selection, crossover, mutation, and accepting the solution. The fitness is used as a measure to rank the chromosomes and to perform parent selection according to the ratio participated by each chromosome in the fitness function in order to reproduce new solutions. However, less fitness members will have also a chance to be selected. Different mechanisms are used to implement the selection step such as the roulette wheel method. The selection will be performed on two chromosomes to reproduce two new chromosomes each time. After selecting the chromosomes, a crossover operation is performed between a

pair of parent chromosomes by selecting a random point in chromosomes and exchanging genes after this point. We choose two random crossing points. The child inherits elements positioned between the two crossover points of the first parent. These elements occupy the same positions, and appear in the same order in the child. The selection and crossover operations may lead to a set of identical chromosomes and the algorithm stops creating new individuals. This may prevent the average fitness improvement and thus trapping into a local optimum. To avoid this problem, a mutation operation is applied where a gene is selected randomly and its value is changed. Mutation performs a larger exploration of the search space, to avoid the premature convergence or the disappearance of the diversity while bringing innovation to the population. The mutation is carried out by reversing the position of two genes. Often, each gene is represented by a bit; the mutation is done by flipping a bit randomly in the chromosome. After crossover and mutation, two new chromosomes are reproduced. Finally, if they are better than their parents, they will be accepted as a new population.

• Stopping Criterion

The stopping criterion is either reaching a maximum number of iterations; either reaching a predefined localization rate (if a rate of k-coverage is ensured, $k = n_{\min}$). Also, we can use a maximum execution time of the algorithm as a stopping criterion.

VI. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of the proposed genetic algorithm in terms of the amount of coverage (coverage rate), the degree of coverage (k-coverage), the number of iterations, and the pareto front. We use the following parameters for the genetic algorithm:

- Area of Simulation ($n \times m$) = 200x300.
- Maximum number of generation = 350.
- Size of population (number of mobile nodes) = $nm / 2\pi r^2$.
- Number of initial stationary nodes = $nm / 2\pi r^2$.
- Probability of mutation = 0.1.
- Probability of crossover = 0.8.
- Number of constraints = 10.

The following figures (Fig.3, Fig.4 and Fig.5) show the difference, in terms of coverage rate between the initial random coverage and the coverage rate ensured by our algorithm.

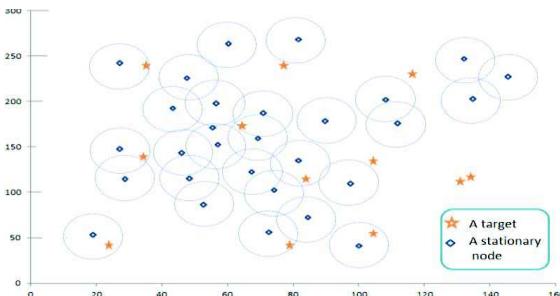


Fig. 3. Initial Random Deployment

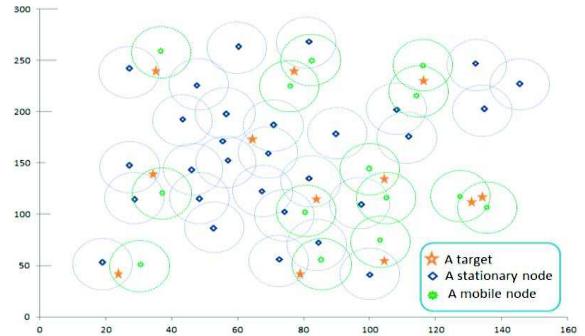


Fig. 4. Deployment after executing the GA for the 2-coverage case

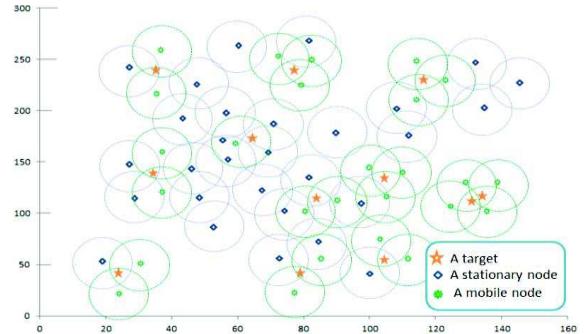


Fig. 5. Deployment after executing the GA for the 3-coverage case

The following Fig.6 represents the coverage rate (axis y) when increasing the number of iterations (axis x). This figure shows that the coverage rate improves when increasing the number of iterations until reaching the demanded degree of coverage.

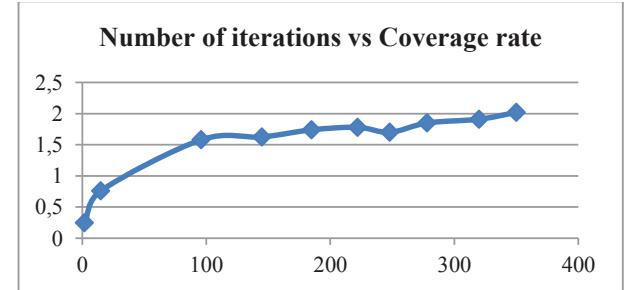


Fig. 6. Number of iterations vs the coverage rate for the 2-coverage case

Actually, our aim is to better locate an acoustic source (target) using genetic algorithm. In fact, as discussed in [8], audio localization performance depends on distance between nodes and the target. In order to perform audio experimentations, we considered an array of two pairs of microphones (as two nodes), two computers, one Smartphone emitting a continuous sound.

Every node is hooked to a computer. We place the Smartphone in an already known position. We then compute for node the angle of arrival of the sound emitted by the Smartphone (as the target). The obtained values of the two angles are automatically stored in order to be used to determine the geographic position of the sound source (Fig.7).

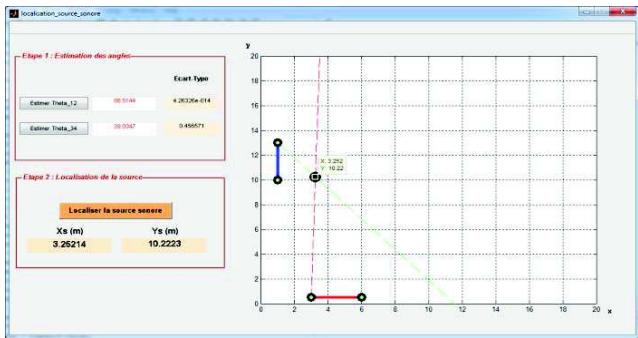


Fig. 7. Screen shot of the developed application for step 2

As we can see in Table I and Fig.8, the error between the estimated positions and the real ones can be explained by the assumptions we had made.

TABLE I. EXPERIMENTAL RESULTS

Real angles (°)		Real positions (m)		Estimated angles (°)		Estimated positions (m)	
θ_{12} theo	θ_{34} theo	X theo	Y theo	θ_{12} esti	θ_{34} esti	X esti	Y esti
90	45	1	1.3	94.15	39.03	0.946	1.25
90	135	1	2.3	95.9	135.048	0.831	2.132
90	90	1	1.8	84.078	89.554	1.13	1.79
45	45	1.4	0.9	44.61	41.77	1.35	0.84
45	90	2.3	1.8	53.28	87.38	1.92	1.735

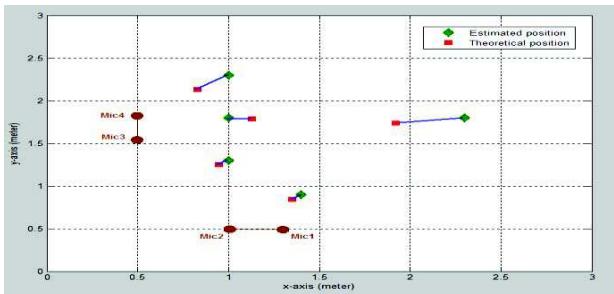


Fig. 8. Acoustic source estimated positions vs theoretical positions

For a multi-objective problem, there is no single solution. The goal of the multi-objective genetic algorithm is to find a set of solutions in that range (ideally with a good spread). The set of solutions is also known as a Pareto front. All solutions on the Pareto front are optimal. In the case of bi-objective problems, informing the decision maker concerning the Pareto front is usually carried out by its visualization. The Fig.9 shows the pareto front of the genetic algorithm. In Fig.9, the x axis represents the values of the first objective function F1 while the axis y represents the values of the second objective function F2.

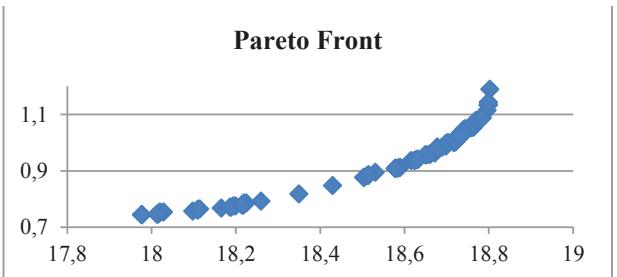


Fig. 9. The Pareto Front of the genetic algorithm

VII. CONCLUSION

In this paper we are interested in deploying a wireless audio-sensor network to optimize coverage and audio localization. We provided a genetic algorithm for an optimized placement of audio-sensor nodes. The aim is to propose an optimal solution for nodes deployment guaranteeing the following objectives: maximizing the coverage area, maximizing the precision audio localization at the level of the detection signal. The proposed genetic algorithm show significant performance improvement in quality compared to the random deployment and the existing methods. As a prospect of our study, we aim to optimize the proposed algorithm in order to ensure the redeployment problem while optimizing different objectives other than the coverage and the localization, such as the lifetime and the network connectivity. Also, we aim to test our contributions by simulation and in reality on a set of testbeds of the OpenWiNo emulator, deployed to the IUT of blagnac in Toulouse.

REFERENCES

- [1] Maulin, P., Chandrasekaran, R. and Venkatesan, S., "Energy Efficient Sensor, Relay and Base Station Placements for Coverage, Connectivity and Routing," IEEE International Performance, Computing and Communications Conference, pp. 581-586, 2005.
- [2] Mnasri S., Nasri N., Val T. An Overview of the deployment paradigms in the Wireless Sensor Networks. International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN 2014), Tunisia – 04-07th November 2014.
- [3] S. Meguerdichian, S. Slijepcevic, V. Karayannidis and M. Potkonjak, "Coverage problems in wireless ad-hoc sensor networks", Proc. IEEE Infocom, vol. 3, pp. 1380-1387, 2001.
- [4] H. Qi, S. S. Iyengar, and K. Chakrabarty, "Multi-resolution data integration using mobile agents in distributed sensor networks", IEEE Transactions on System, Man and Cybernetics (Part C), vol. 31, pp. 383-391, August 2001.
- [5] R. R. Brooks and S. S. Iyengar, Multi-Sensor Fusion: Fundamentals and Applications with Software, Prentice Hall, Upper Saddle River, NJ, 1997.
- [6] A. Howard, M. J. Matarić and G. S. Sukhatme, "Mobile Sensor Network Deployment Using Potential Field: a distributed scalable solution to the area coverage problem", to appear in Proc. International Conference on Distributed Autonomous Robotic Systems", June 2002.
- [7] A. Konstantinidis, K. Yang, Q. Zhang, and D. Zeinalipour-Yazti, "A multi-objective evolutionary algorithm for the deployment and power assignment problem in wireless sensor networks," Computer Networks 54(6): 960-976, 2010.
- [8] Thaljaoui A., Val T., Brulin D. and Nasri N., "Real Time Acoustic Localization of Elderly Persons in a Smart Home," The International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN2014), 04-07th November, Sousse, Tunisia 2014.