New Fuzzy CSP for an Optimized Mobile Robot's Path Tracking using Genetic Algorithms

Tlijani hayet
LARA : departement Génie Electrique
Ecole Nationale des Ingénieurs de Tunis (ENIT)
Tunis, Tunisie
tlijanihayet {at} live.fr

K M'sirdi Nacer, Knani Jilani
LSIS: departement Informatique Industrielle, LARA: departement Génie Electrique
Polytechnique Marseille, ENIT
Marseille, France; Tunis, Tunisie

Abstract– **Even though, the problems of optimization are considered as a type of a constraint satisfaction problem: COP (Constraint Optimization problem), we can study other constraint such as temporal one. In this frame goes our first work which is interested in the optimization of the navigation of a mobile robot under a time window. The studied robot will be used for the surveillance (Surveyor Robot). Indeed, in the previous work we have focused on the minimization of the length of this trajectory by using the approach of the genetic algorithms. At the level of our present researches we shall consider the time delayed in the passage between one surveyed site and the following one. So that we process to model the problem as a satisfaction constraint problem. Consequently, a new algorithm for the optimization will be feigned using genetic algorithms and guided by the techniques of CSP (Constraint Satisfaction Problem). The new algorithm allows minimizing the length of a trajectory crossed by a mobile robot of surveillance and respecting the time fixed to take during the passage through two successive sites. After that, this work will be improved using the paradigms fuzzy logic. In the following work, we will present a new form to the Fuzzy CSP to propose a fuzzy controller for a two-wheeled mobile robot moves under a time window. In dead the fuzzy controller controls the acceleration. It aims to give the efficient acceleration so that the controlled robot can move from one position to another one in a delimited time window. Furthermore, it converges the time delay between positions in the path tracking as closer as possible to this time window.**

Keywords-Mobile Robot; Path Tracking; Optimization; Genetic algorithms; Constraint Satisfaction Problem; Fuzzy Logic

I. INTRODUCTION

Many engineering problems in practice are of combinatorial optimization type that can be modeled as constraint satisfaction problems (CSPs). Our research deals with an example of this type of problem. It concerns the minimization of a path tracking for a mobile robot of surveillance (a surveyor robot) moving under temporal constraint. The temporal constraint consists of a time window that limits the time delayed in the passage between one surveyed site and the following one across the path tracking. In an early work we have focused on the minimization of the path tracking [1] using the genetic algorithms. The consideration of temporal aspects requires a modeling of this problem with a satisfaction of constraints. So that, a hybridization of the algorithm of optimization already developed by the approach of CSP, is indispensable for our case.

The new algorithm is a hybrid algorithm allowing minimizing the length of a trajectory crossed by the mobile robot and respecting the time window between two successive sites. The heuristics used to guide the solver of this problem led to change some of genetic algorithms' operators. The crossover was changed to verify the temporal constraint. However the operator of mutation was conserved as it was used in the previous works. In dead, the definition of the time constraint was integrated and resolved in the crossover operator of the generated algorithm.

The navigation under constraints sometimes engenders ambiguous solutions. This urges us to think of a strong strategy and intelligent as the fuzzy logic. This approach of artificial intelligence was used for the command of a small robot, to insure the follow-up of a complex and pre-calculated trajectory. In this work, a controller based on the fuzzy logic is adopted on the model of the two-wheeled mobile robot described in [2]. The objective of this fuzzy controller is to respect as close as possible the time window. It has to correct the velocity calculated according to the solver that respects the time window between two successive positions in the path tracking. Then, we notice that we can improve the result by making the time delay closer to the value of the time window. Our idea was how we can change the parameters that impacted the navigation of the mobile robot. To answer this, we have centered on the acceleration. Besides, we have enhanced our solution by the paradigm of the fuzzy logic to make the results more natural and efficient.

This paper will be organized as following. Firstly, we describe the problem statement in section 2. This section specifies the constraint satisfaction problem related to our case and the fuzzy system developed. Secondly, we define in section 3, the heuristics and the generated algorithm to solve the problem of temporal constraint. Furthermore, we illustrate the new algorithm based on genetic algorithms combined with CSP technique and fuzzy logic method. We end this work by the section 4 that gives the results of the simulation of the executed plan.

II. PROBLEM DESCRIPTION

Commonly, the heuristics adapted to deal with time constraint build a graph of navigation. It corresponds to the projection of states on the facts relative to the position of the vehicle. The costs of the arcs of this graph in terms of time or length are calculated by a planner of road: a graph of planning is built and used to estimate the consumption of time or resource necessary for the reaching of every objective [3].

In the present work we have built a graph to characterize the movement of a mobile robot in its workspace. Then we use a planner of road that estimates a reference trajectory for the studied mobile robot. The cost estimated for this trajectory corresponds to the crossed distance. A heuristics of optimization is used to have a minimal length for this trajectory. It was the genetic algorithms approach that we have carried out in a previous work [1].

The graph used by the planner for the generated reference path connects two objectives. Each objective corresponds in reality to the various artificial beacons marking the map of the workspace of our mobile robot. The passage of an objective (position) in the other one is restricted with a Time Windows.

Consequently, our problem converges in a definition of a Constraint Satisfaction Problem (CSP). Further than the genetic algorithms that minimize the reference trajectory, we have to use the techniques of the CSP. Then a Genetic Algorithms-based plan is examined where a path solution obtained with the combination between Genetic Algorithms and Constraint Satisfaction Problem techniques.

The different beacons that structure the path solution will be the data source to a fuzzy controller realized to improve the time delay between two positions of the path tracking. In each position, the current velocity of the controlled robot is an input for the fuzzy controller. The second input is the time delay between the current position and the next sub goal to reach [4].

So that we combine three approach: the genetic algorithms and the CSP techniques to resolve the problematic that minimize the distance crossed along a reference trajectory of a mobile robot constrained with a time windows. Besides, the fuzzy logic to realize a fuzzy controller that adjusts the real time delays as close as possible to the imposed time window.

The presented work deals with the hybridization between the genetic algorithms, techniques of Constrain Satisfaction Problem (CSP) and fuzzy logic to solve the temporal constraint.

A. *The variables of the problem*

The graph used by the generated planner is a set of n knots corresponding to the artificial beacons' positions in the configuration space of the mobile robot. These beacons mark the sites to survey by the studied mobile robot. Two successive knots k and k+1 represent an arc of the graph that defines the movement of the mobile robot from an objective to the other one. On one hand, k+1 is the knot to which the vehicle goes and on the other hand, k and k+1 describe the respective postures of the mobile robot (vehicle).

Thus at a fixed moment the state is defined by:

- The position of artificial beacon to which the mobile robot is going.
- t_k : Time in the position k;
- t_{k+1} : Time in the position k+1;
- The posture of the mobile robot (position and speed): described in the section 3 of this paper.
- T : Time between two successive positions: it determines the time windows.
- d_{ij} = Distance between the artificial beacons.
- n = Number of the artificial beacons in the space of work.
- k = Current position of the mobile robot; k=1,..,n.
- xij = Binary Variable $\begin{cases} = 1 \text{ if the beacon (surveyed site) j is visited immediately after the beacon i ;} \\ = 0 \text{ else.} \end{cases}$

B. *Formalization of the problem*

Choosing the right model and choosing the right constraint satisfaction algorithm is crucial for efficient solving of the problem. We suggest in this section a formalism that allows describing completely the problem. In one hand, this formalism generates a solver for the problem of optimization of the mobile robot path tracking with time window time. In the other hand it defines the temporal constraint with fuzzy logic paradigms.

1) Mathematical formulation: The problem consists in minimizing the length of the path tracking crossed under a time window. First of all we give the objective function is:

- Minimize:

$$D = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij} x_{ij} \qquad (1)$$

- Constraints:

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad (2)$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad (3)$$

$$t_{k+1} - t_k \leq T \qquad (4)$$

The equation (2) defines the constraint which makes sure that we go out only once of each beacons (surveyed sites). Whereas the equation (3) describes the constraint which verifies that we enter only once every surveyed site. These constraints allow visiting all the sites only once. The temporal constraint is illustrated by the equation (4) where T the period of the time window; and t_k and t_{k+1} the moments of passage of the mobile robot respectively by the position k and the one k +1.

2) CSP definition: Reference [5] defined a constraint satisfaction problem by a triple (X,D,Ci), where X is a set of variables, D is a domain of values, and C is a set of constraints C1(S1)..Cn (Sn) where each Si is a set of variables. A constraint Ci is a combination of valid values for the variables Si. A solution to the CSP is an assignment of values to S1...Sn that satisfies all constraints.

In our case the triple is defined as following:

- X = {(xb_i ,yb_i) / position of a surveyed site marked by an artificial beacon ; i =1,………,n},
- D= {bi / bi ϵ N: number of each beacon in the path},
- C= {equation (2), equation (3), equation (4)}.

The equations (2), (3) and (4) form the constraints imposed on the movement of the mobile robot in its configuration space.

3) Fuzzy controller Model: Fuzzy CSP is the classical Constraint Satisfaction Problem which has been extended to incorporate different fuzzy logic. The present application of this paradigm is a new way to the use of fuzzy logic in the resolved constrained problem. Because in our work, we use the fuzzification to give results according to the range of values that satisfy the temporal constraint defined by equation (4).

The fuzzy system to realize is a two-input-and-one-output fuzzy system. The two inputs to the fuzzy system are: the velocity of the mobile robot in the position of the current beacon and the time needed to reach the next goal. The output corresponds to the acceleration so that the mobile robot reaches the next surveyed site in a time close to the time window. The described fuzzy system is illustrated in Fig . 1.

a) *Fuzzification :* We propose the description of the velocity, time and acceleration as following: the Big Negative (BN), the Medium Negative (MN), the Small Negative (SN), the Zero (Z), Small Positive (SP), Medium Positive (PM), Big Positive (BP)), the Big (B), the Very Big (VG), the Small (S), the very Small (VS), the Very Low (TF), the Low (F). These linguistic terms, are represented by fuzzy sets partitioned into crisp universes.

To describe the functions member of fuzzy sets variables of inputs and output, we have used triangular membership (see fig. 2) indicates. Different membership functions such as Gaussian, trapezoidal, and bell could have also been used. Each one of these membership functions has its own effects on the fuzzy logic controller output [6]. However, triangular membership functions are more convenient for expressing the concept because it is easier to intercept membership degrees from a triangle.

The equation (5) is used to represent the fuzzy triangular membership functions.

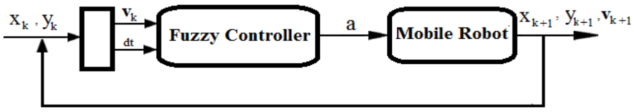

Figure 1. Diagram block of the fuzzy controller system for a two-wheeled mobile robot

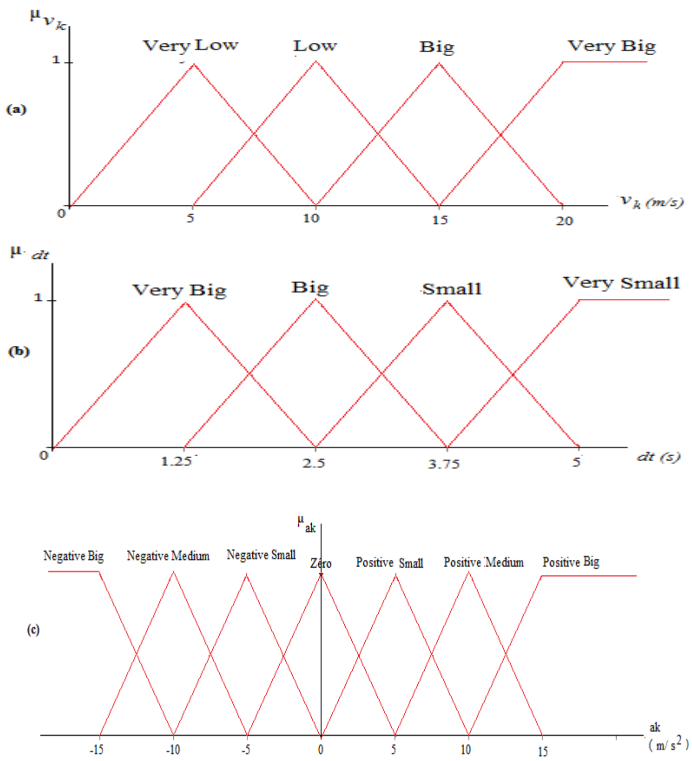

Figure 2. Function of membership: (a) fuzzy subsets of the velocity v, (b) fuzzy subsets of the dt, and (c) the fuzzy subsets of the acceleration a.

$$\mu(x) = \max\left[\min\left(\frac{x-x_1}{x_2-x_1}, \frac{x_3-x}{x_3-x_2}\right), 0\right] \quad (5)$$

Where, x is the crisp values from one of the three universes v, dt, and a. x1 is the left end point on the corresponding crisp universe as the x2 and x3 are the crisp points corresponding peak and right end points, respectively.

b) *Inference:* The fuzzy decision table represents the fuzzy rules base. It defines the knowledge and abilities of a human operator who makes necessary adjustments to operate the system with needed velocity and to respect the time window. The output from the fuzzy logic controller, which is the acceleration, models the actions that a human operator would decide: whether the velocity of the mobile robot is to be increased or decreased according to the time remained to reach the next beacon.

A rule decision table can be formed as shown in TABLE I. This table is built manually.

TABLE I. Fuzzy Base rule of the fuzzy controller system for the mobile robot

a		v_k			
		VL	***L***	***B***	***VB***
dt	***VB***	PB	PB	PM	PS
	B	PB	PM	PS	Z
	S	PM	PS	NS	NM
	VS	PS	Z	NB	NB

The mode of use of this table is illustrated by the fuzzy rules proposed by the system described as follows:

- R1: if (((dt is VB) and ((v_k is VL) or (v_k is L)) or ((dt is B) and (v_k is VL)) Then (a is Positive Big);
- R2: if (((dt is VB) and (v_k is L)) or ((dt is VB) and (v_k is B))) Then (a is Positive Medium);
- R3: if (((dt is B)) and (v_k is B)) or ((dt is VB)) and (v_k is VB)) or ((dt is S) and (v_k is L)) or ((dt is very S) and (v_k is VL))) Then (a is Positive Small);
- R4: if (((dt is B)) and (v_k is VB))) or ((dt is VS) and (v_k is L))) Then (a is Zero);
- R5: if ((dt is S) and ((v_k is B)) Then (a has is Negative Small) ;
- R6: if ((dt is S) and ((v_k is VB)) Then (a is Negative Medium));
- R7: if ((dt is very S) and ((v_k is VB) or (v_k is B)) Then (a is Negative Big);

These seven rules will form the base rule of the studied fuzzy controller. We notice that in these rules we use of logical operators "AND" and "OR". The realization of these logical operations connected to the method of used inference. In our case we shall use the method Max-Min [7]. This method of inference realizes the conditions connected by OR using the maximum whereas the conditions connected by AND using the minimum.

c) *Defuzzification:* The defuzzification is the step coming after specifying the membership functions of variables (fuzzification) and establishing the base of rules (inference). It allows converting the linguistic variables of the fuzzy controller into the physical variables. Based on the weighted average method, the final outputs of the controller fuzzy system can be described by equation (6):

$$a = \frac{\sum_{i=1}^{7} \mu_{a(\alpha_i)} {}^{*}\alpha_i}{\sum_{i=1}^{7} \mu_a(\alpha_i)} \quad (6)$$

4) Localization and the posture of the mobile robot: In [2] we have specified the model of the used mobile robot. It's a two-wheeled mobile robot is considered and its structure is described (see Fig 3), where X-Y is the global coordinates and xm-ym is the local coordinates which is fixed to the robot with its center p as the origin. As shown in the latest figure, the body of the mobile robot is of symmetric shape and the center of mass is at the geometric center p of the body. R is the radius of the wheel and L is the displacement from the center of robot to the center of wheel. The set (x_o, y_o) represents the position of the geometric center p in the world X-Y coordinates, and the angle θ indicates the orientation of the robot. The angle θ is taken counter clockwise from the X-axis to the x_m-axis.

The two wheels are fixed and controlled independently by two motors. For this reason, the considered wheeled robot is equipped with motors that are driven by on board computer. It is assumed that the wheeled mobile robot is made of a rigid frame, non-deformable wheels that do not slip, and it is moving on a horizontal plane. We assume this mobile robot to a one particle in its configuration space.

According to this description, and the mobile robot localization using artificial landmarks studied in [9] [10] we can define the posture of the used mobile robot by the following equation (7):

$$X_{k+1} = \begin{Bmatrix} x_k + v_k \cos\theta \, dt \\ y_k + v_k \sin\theta \, dt \\ v_{k+1} \\ \theta_k + \dot{\theta}_k \, dt \\ \dot{\theta}_{k+1} \end{Bmatrix} \quad (7)$$

Where:

- X_{k+1}: the vector of the posture at the state k+1 generated in function of the previous state;
- (x_k,y_k): the position of the center of the mobile robot at the state k;
- v_{k+1} : The linear velocity of the mobile robot at the state k+1;
- θ_k : The angular position of the center of the mobile robot at the state k;
- $\dot{\theta}_{k+1}$: The angular velocity of the mobile robot at the state k+1;

This vector describes the posture of the mobile robot in its configuration space referring to its previous one. So that we can specify at any time the different coordinates of the studied mobile robot. In dead, we will use this vector after in the algorithm of the crossover operator for the genetic algorithm, to calculate the position that respects the temporal constraint. Furthermore, it's used to determine the velocity in each beacon across the path tracking for the fuzzy controller.

The linear velocity is defined by the equation (8):

$$v_{k+1} = v_k + a\,dt \quad (8)$$

Where:

- v_k , v_{k+1} : Linear velocity in the position k, k+1 respectively ;
- a : Acceleration needed so that dt will be closer to the value of the time window;
- dt : Time delay between two positions of the mobile robot (k+1 and k);

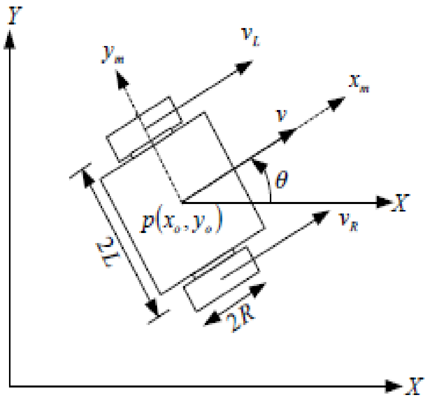

Figure 3. Posture definition of a two wheeled mobile robot

III. USED HEURISTICS AND GENERATED ALGORITHMS

A. The genetic algorithms operators

Genetic algorithm is particularly easy to implement and promises substantial gains in performance. Its basic idea is to keep several sub-populations that are processed by genetic algorithm. Furthermore, a migration mechanism produces a chromosome exchange between sub-population. The basic algorithm that gives a minimized path tracking was studied and tested in [1].

The main characteristics of the proposed GA consist of the following:

- A string of number encoding a feasible path joining all detected beacons in the work-space.
- A gene in the chromosome is an artificial beacon in the configuration space of the mobile robot.
- Each path is an individual that has a length.
- The length of the path is a fitness degree of the individual.
- Selection: the individual chromosomes are selected for reproduction according to the elitist selection: the individual having the best fitness is chosen.
- Crossover: new populations of chromosomes are generated using the algorithm of recombination based on KFP crossover [11].
- Mutation: a perturbation applied to the individuals by reversing the path between two positions.

B. The algorithm of the crossover using KFP operators

The KFP crossover is an operator of recombination Elaborated by Karoly F. Pale in 1993 [11]. It was introduced a heuristics based on a principle similar to that of a selection by roulette. This operator creates an individual son by beginning with a gene (position in the path tracking) chosen at random, then by including the following genes according to a probability defined according to the distance between the current one and the following in each of both parents.

At the first solution we adopted this algorithm; as it's shown below in the algorithm of Fig .4 the probability (pa and pb) are calculated according to the distance:

d_a = Distance (iCurrentGnum, a);
d_b = Distance (iCurrentGnum, b);
Where:
d_a : distance between the current position in the individual son and the gene a in the first parent.
d_b: distance between the current position in the individual son and the gene b in the second parent.
Distance: a function developed that returns the distance between two positions in the search space.

The individuals correspond in our work to the possible paths connecting the artificial beacons which mark the configuration space of the mobile robot. These beacons form the genes of the individual. The genes are coded by means of numbers (integer type). The elaborated algorithm of crossover is presented in Fig .4.

C. New crossover operator based on KFP crossover and Neighbourhood method

Meta-heuristics are methods that combine local improvement procedures and higher level strategies to create a process capable of escaping from local optima and performing a robust search for a solution space [12]. Nowadays, meta-heuristics can be seen as intelligent strategies to design or improve heuristics procedures with a high performance, generally combining constructive methods, local search methods, concepts that come from Artificial Intelligence, biological evolution and statistics methods. In this part of our research, we have analyzed the hybridization of the CSP resolving the problem of temporal constraint and Genetic algorithms as representative methods. Our proposed meta-heuristic consist of combining the heuristic of neighborhood and the KFP crossover [8] [11].

The methods of neighborhood are based on the notion of neighborhood. In [13] is defined as following:

Let X the set of the admissible configuration of a problem:

- The neighborhood is an application N: $X \rightarrow 2^X$.
- The mechanism of the neighborhood exploration is a procedure that defines how to get a configuration $s' \in N(s)$ from a configuration $s \in X$. A configuration s is a local optima (minima) of the neighborhood N if f(s) <= f (s') $\forall s' \in N(s)$ where f is the objective function.

A typical method of neighborhood begins with an initial configuration, and realizes then an iterative process which consists in replacing the current configuration by one of its neighbors in the objective function. This process stops and returns the best configuration found when the ending condition is realized. This condition concerns generally a limit for the number of iterations or an objective to realize.

We have opted for the method of neighborhood to look for the gene which verifies the temporal constraint the ending condition was fixed to the number of neighbor to visit and verify. In Fig. 6 we illustrate the algorithm where we have used this method to resolve the temporal constraint of equation (4).

The algorithm of Fig. 6 is based on neighborhood. It's called after every crossover to test the individual son. The crossover algorithm used is shown in Fig .4. It was developed and tested in the previous work [2] based on KFP algorithm [11]. As results, the new crossover algorithm shown in Fig. 5 has two steps: first one calls KFP crossover, then the second calls CSPverif illustrated in Fig. 6 This assures that the individual obtained from the process of recombination, respects the temporal constraint defined in the equation (4). As result we generate a solver to the optimization problem with time window. The new crossover operator is called CrossoverCSP.

```
Procedure CrossoverKFP (Individual &I1, Individual &I2,
                        Individual &I_temp)
 Variable
   i, iCurrentGnum, iCurrentPos, a, b: integer
   d_a, d_b, p_a, p_b: float
 Begin
   For (i=0 to NbG) do
     I_temp.Amers[i].num = -1
   End for
  iCurrentGnum ← NbG * (rand () / (RAND_MAX + 1.0)
  I_temp.Amers[0].num ← iCurrentGnum
  For (iCurrentPos=1 to NbG) do
{Look for the next gene after iCurrentAmers}
     a ← I1.FindGPos (iCurrentGnum);
     b ← I2.FindGPos (iCurrentGnum);
{Gene from parent1: I1}
    Repeat
     a = (a + 1) mod NbG
    Until (I_temp.FindGPos(I1.Amers[a].num) = -1)
    a ← I1.Amers[a].num
{Gene from parent2: I2}
    Repeat
     b = (b + 1) mod NbG
    Until (I_temp.FindGPos(I2.Amers[b].num) = -1)
    b = I2.Amers[b].num
{Calculus the distances and probabilities for a and b}
     d_a = Distance (iCurrentGnum, a)
     d_b = Distance (iCurrentGnum, b)
     p_a = d_a * (rand () / (RAND_MAX + 1.0)
     p_b = d_b * (rand () / (RAND_MAX + 1.0)
    If (p_a < p_b) then
     iCurrentGnum ← a
    Else
     iCurrentGnum ← b
    End if
{Insert the new gene to the individual son I_temp}
    I_temp.Insert ( iCurrentGnum)
 End for
{Evaluation of the individual son I_temp}
 I_temp.Eval()
End {CrossoverKFP}
```

Figure 4. Algorithm of the crossover based on KFP operators

```
Procedure CorssoverCSP ( int T, Individual &I1, Individual &I2,
                         Individual &I_temp )

  Begin {CorssoverCSP}

     CrossoverKFP ( I1, I2, I_temp)

     CSPverif ( T, I_temp)

  End {CorssoverCSP}
```

Figure 5. Algorithm of the new crossover

```
Procedure CSPverif(int T, Individual &I)
  Variable
     l,m,n : int
     t,d,D : float
Begin
  t←T
  For (i=0 to NbG) do
     d=I.Distance(i,i+1)
{calcul the position that respecte time window T}
     x=I.Amers[i].AmersPosX + vl*cos(a)*t;
     y=I.Amers[i].AmersPosY+ vl*sin(a)*t
     a=a+va*t
     D=(sqrt(((x-I.Amers[i].AmersPosX)*
     (x-I.Amers[i].AmersPosX))+
     ((y-I.Amers[i].AmersPosY)*
     (y-I.Amers[i].AmersPosY))));
     m=i+1
     if (d>D) then
{test the 3 neighbours for each beacons (gene)}
     For (j=0 to 2) do
         l←FindNeighbour(i,j)
         n←I.FindGPos(l)
{calcul the distance between ith gene the new
neighbour}
     d←I.Distance(i,n)
     If (d<D) then
        m=n;{temporal constraint is verified}
     End if
{mutation of individual I exchange the gene i+1 by
the gene m }
     I.Mutate(i+1,m)
     {update d}
     d←I.Distance(i,i+1)
     end For
{update the linear velocity of the mobile robot in
the new position :gene}
  t= d/vl
  vl=vl+acc*t {acc is a global variable indicates the
              value of the acceleration}
  End for
End {CSPVerif}
```

Figure 6. Algorithm of CSP verification based on neighbourhood heuristic

D. New Genetic Algorithm

Genetic programming has in recent times been verified to be a practical approach for mobile robot control and navigation [10]. In this context, we address the design of fuzzy logic controllers handling temporal constraint in a Genetic Algorithms-based plan.

The new algorithm is a combination between the genetic algorithms, the approach of Constraint Satisfaction Problem and the approach of the fuzzy logic.

The execution of this new algorithm generates the value of the necessary acceleration in every position of the optimized path. This later yield from the hybridization of the genetic algorithms and Constraint Satisfaction Problem techniques.

In fact, the evolutionary approach gives the shortest path to go and the Constraint Satisfaction Problem techniques bound the passage way between two successive beacons by a time window. By using the fuzzy logic, we correct the time delayed between every two consecutive beacons in the path tracking.

Consequently, the result is a minimal path tracking for a mobile robot respecting as much as possible a time window. The new algorithm based on genetic algorithms is described by the organigram in Fig. 7.

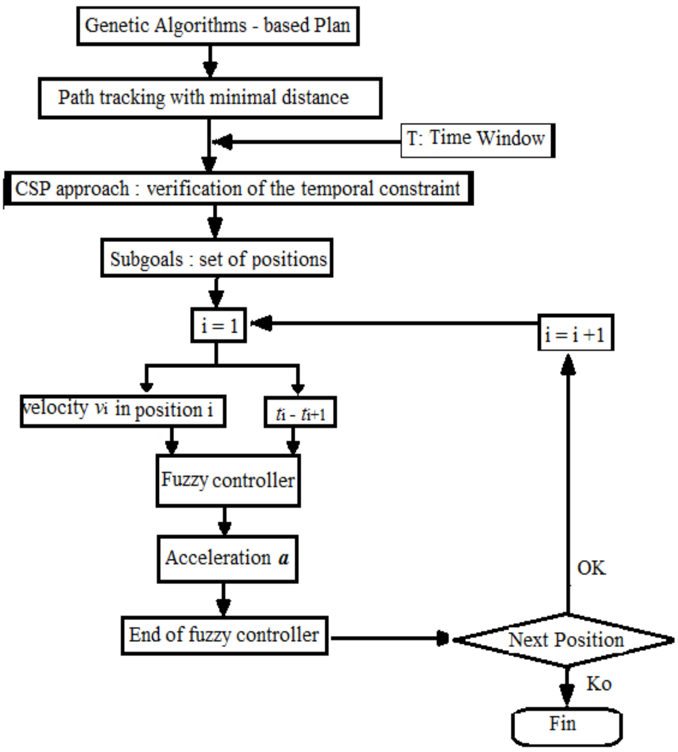

Figure 7. Organigram: combination between the genetic algorithms, Constraint Satisfaction Problem approach and fuzzy logic

IV. Simulation and results

The simulation of the new genetic algorithm and the different functions related to the algorithms illustrated in Fig. 4, Fig .5 and Fig. 6, are realized by using C ++ under the development environment of the Microsoft Visual Studio 2010.

We assume that the mobile robot is a one particle in its configuration space. It's a horizontal plan equipped by a number of artificial beacons (n = 5). Each one represents anobjective to reach which is a site to survey. The positions of the artificial beacons are represented in the TABLE II.

Linear speeds and angular initials are respectively 2m/s and 0.5rad/s. The temporal window has for value 5mn and the number of neighborhood is fixed to 3. The time window has for value T=5mn and the number of neighbors is fixed to 3.

The execution begins with the initial population described by the TABLE III. The evolution of the various generations in this case is described by TABLE IV, TABLE V and TABLE VI. According to the values obtained in the final generation (TABLE VI), we can deduct that the solution is the path coded by the successive positions 04231 having a length of 12.6084.

The evolution of the individuals resulting from every generation is illustrated in Fig. 8. Then the Fig. 9 represents the path tracking solution of the generated algorithm which is encoded by the integer sequence: 04321.

Table II COORDINATED BY THE POSITIONS OF THE ARTIFICIAL BEACONS

Artificial beacon's code	0	1	2	3	4
X	7.0	9.0	5.0	3.0	5.0
Y	8.0	1.0	2.0	4.0	6.0

Table III 1ST GENERATION

Individual Code	04321	01423	02413	21430	30214
Fitness	12.6084	20.5117	23.4359	19.0115	22.5076

Table IV 2ND GENERATION

Individual Code	04321	01432	02413	21430	30214
Fitness	12.6084	19.3401	23.4359	19.0115	22.5076

Table V 3ND GENERATION

Individual Code	04321	01342	02413	21430	30214
Fitness	12.6084	20.8167	23.4359	19.0115	22.5076

Table VI FINAL GENERATION

Individual Code	04321	01234	02413	21430	30214
Fitness	12.6084	18.2316	23.4359	19.0115	22.5076

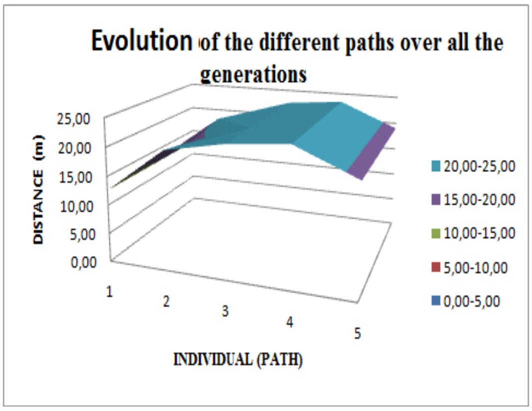

Figure 8. Evolution of different individuals of the new genetic algorithm using genetic algorithms and CSP techniques

The definition of the linguistic variables inputs and output: dt, v_k and a, are defined in a class called: RM_En_Flou. In this class we define the various membership functions of the linguistic variables as member functions of this class. The inference is concretized by a class that we have conscript RB_Moteur_Inference. In this class we have defines an array corresponding to the inference table (Table I). Among the member functions in the class of inference, we have program the function of inference Max-Min defined in equation 3 and the function used for the defuzzification which bases on the equation 4.

The values of the acceleration calculated by the fuzzy controller for the deferent points of the trajectory are recapitulated in the following TABLE VII. The trajectory is yielded by the new algorithm that combines the approach of the genetic algorithms and the techniques of the Constraint Satisfaction Problem.

Points studied in the trajectory are 0, 4, 3, and 2; we notice the absence of the point 1 of this trajectory because this point marks the end and as a consequence it hasn't successor.

The next Fig. 10 represents respectively the variation of the velocity and the acceleration along the resultant path tracking (04321).

TABLE VII The whole decision by using the fuzzy logic paradigms

Code position	**0**	**4**	**3**	**2**
V_k (m/s)	2	20	13.44	67
dt (mn)	3	2.75	2	4.3
a (m/s^2)	6	-1.14	6.9	-9.7

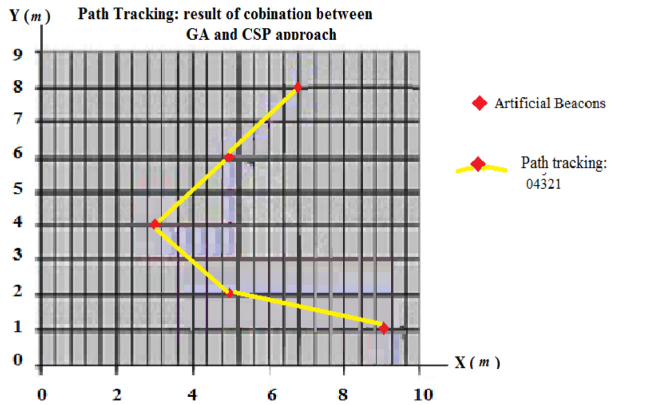

Figure 9. Path tracking: result of the hybrid algorithm combining genetic algorithms and CSP techniques

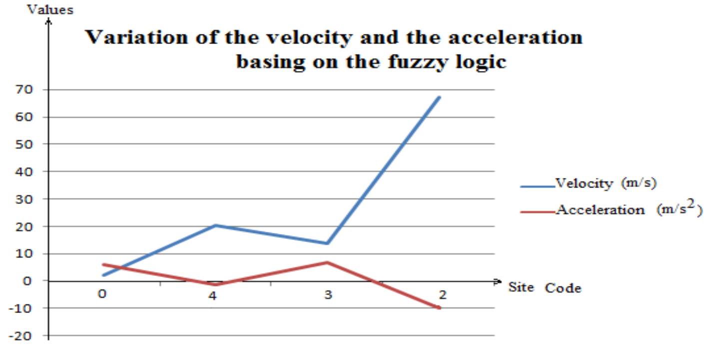

Figure 10. Variation of the speed and the acceleration of the Mobile Robot by basing itself on the vague logic along the trajectory crossing sites to be watched and respecting a temporal window.

V. CONCLUSION

The optimization of the mobile robot track with time windows is a combinatorial optimization type that can be modeled as constraint satisfaction problems. For this reason we have used hybridization between the genetic algorithms and the CSP techniques to have a solver to the temporal constraint. A meta-heuristics was also defined that combines the genetic algorithm and neighborhood method to guide the solver to the solution.

Actually, the mobile robot has to satisfy several requirements during its navigation. It has to make reliable decision. Executing an optimal plan under temporal constraint seems a typical example. In this purpose we must specify suitably the parameters of the movement of the mobile robot: the acceleration and the velocity. These arbitrations were insured by some control approach such as the genetic algorithms, the techniques of Constraint Satisfaction Problem and the fuzzy logic. The hybridization combining these three approaches, shows a new way to treat a constraint defined on a system with fuzzy logic. Actually, this new method approve that the solver can be guided by human preferences. So that we acquire human abilities to the mobile robot for its decisions in navigation.

REFRENCES

[1] Hayet TLIJANI, Jilani KNANI, "Optimization of the mobile robot's track using Genetic algorithm", In10th International conference on Sciences and Techniques of Automatic control & computer engineering – STA'2009, Hammamet-Tunisia. (December 2009)

[2] Hayet Tlijani, Nacer K.M'sirdi, .Jilani Knani, "Optimization of the mobile robot's Path Tracking: Combination of Genetic algorithm, Artificial Potential Field and Backtracking Algorithm", International Conference on Communications, Computing and Control Applications – CCCA'11, Hammamet- Tunisia, March 2011.

[3] Reshef Meir, Ariel D. Procaccia, and Jeffrey S. Rosenschein, "Strategyproof Classification Under Constant Hypotheses: A Tale of Two Functions ", In AAAI-08: Proc. 23rd AAAI Conference on Artificial Intelligence, pp. 126-131, Jul 2008.

[4] Tlijani Hayet, Tlijani Hatem, Knani Jilani, "Navigation control of a mobile robot under time constraint using the fuzzy logic", The International Conference on Control, Engineering & Information Technology (*CEIT'14)*, in press.

[5] LASSOUED Yassine, "Etude de paramétrage d'un algorithme d'optimisation hybride", France, 2008.

[6] Altas I. H, "The Effects of Fuzziness in Fuzzy Logic Controllers", 2nd International Symposium on Intelligent Manufacturing Systems, Sakarya – Turkey, 1998.

[7] Muhammad Siddique, "Fuzzy decision making using max-min and MMR methods", 03, pp. 23. ING/School of Engineering, August 2009.

[8] H. Tlijani, J. Knani, K.N. M'sirdi, "Optimization of a mobile robot's path tracking with time windows using Hybridization of genetic algorithms and CSP approach", International Conference on METAheuristics and Natural Inspired Computing –META'12, Sousse -Tunisia, October 2012.

[9] Glover F., "Future paths for integer programming and links to artificial intelligence", Computers and operations Research, 13, p 533-549, 1986.

[10] K.F. Pál., "Genetic algorithms for the travelling salesman problem based on a heuristic crossover operation", Institute of Nuclear Research of the Hungarian, Academy of Sciences. Debrecen, Hungary, 1993.

[11] H.Tlijani, J.Knani, Nacer. K. M'sirdi, "Hybrid Algorithm based on Genetic Algorithms and CSP approach to plane an Optimized Path Tracking for a Mobile Robot moving with Time Windows ", IOSR-JCE, in press

[12] P. H. Mills, "Extensions to Guided Local Search", A thesis submitted for the degree of Ph.D. Department of Computer Science University of Essex, 2002.

[13] Jin-Kao Hao, Philippe Galinier, Michel Habib, "Méthaheuristiques pour l'optimisation combinatoire et l'affectation sous contraintes", Revue d'Intelligence Artificielle, 1999.