Proceedings of the 2011 IEEE
International Conference on Robotics and Biomimetics
December 7-11, 2011, Phuket, Thailand

Multi-Objective Path Planning using Spline Representation

Faez Ahmed and Kalyanmoy Deb

***Abstract*— Off-line point to point navigation to calculate feasible paths and optimize them for different objectives is computationally difficult. Path planning problem is truly a multi-objective problem, as reaching the goal point in short time is desirable for an autonomous vehicle while ability to generate safe paths in crucial for vehicle viability. Path representation methodologies using piecewise polynomial and B-splines have been used to ensure smooth paths. Multi-objective path planning studies using NSGA-II algorithm to optimize path length and safety measures computed using one of the three metrics (i) an artificial potential field, (ii) extent of obstacle hindrance and (iii) a measure of visibility are implemented. Multiple trade-off solutions are obtained on complex scenarios. The results indicate the usefulness of treating path planning as a multi-objective problem.**

I. INTRODUCTION

In recent years we have seen much advancement in the field of Industrial Robotics but mobile intelligent machines have mainly been confined to research labs. For an autonomous vehicle to be used in real world applications it must be able to navigate autonomously and take intelligent decisions. Vehicle path planning comprises of not only generating collision free paths from a given location to its destination point but also finding the optimal path. [1] and [2] provides broad coverage of the field of motion planning algorithms focusing on vehicle motion planning. Most path planning studies done in literature concentrate on finding the shortest path to destination though in practical mobile robot applications it should consider multiple objectives like path safety,path smoothness and visibility. Considering path safety as an objective would ensure that the resultant paths are navigable and do not pose high risk to robot safety. Safety should not be defined just as collision free path but it should also take into account surrounding effects. Hence we have proposed different measures of safety which are useful in varied application scenarios. A multi-objective study for path length and safety using binary genes has been done in [3]. In a bi-objective genetic optimization for path length and safety the resultant paths may come out to be zig-zag with sharp turns which a robot may not be able traverse. Similarly a solution path passing through narrow lanes will have low visibility and obstacle detection sensors on board will have lower probability of detecting a possible danger. The robot would prefer to move through areas where its sensor field of view is maximized ensuring less chances of possible collision. While path length and safety are the major objectives of optimization, smoothness of path should be implicit in the path representation scheme and should not be taken as an independent objective.

Faez Ahmed is a Graduate student at Department of Mechanical Engineering, IIT Kanpur, Kanpur, India faez@iitk.ac.in
Kalyanmoy Deb is a Professor at Department of Mechanical Engineering, IIT Kanpur, Kanpur, India deb@iitk.ac.in

We propose a bi-objective optimization for path length and safety using a Genetic Algorithm (GA) with a path representation method using B-splines which inherently ensure path smoothness. Piecewise polynomial and B-splines representation schemes have been used. Three different forms of safety measure have been defined. First uses path potential field approach where operator can customize the difficulty level of environment using an artificial potential field. Second approach of obstacle hindrance is operator independent where robot tries to avoid obstacle-cluttered areas by minimizing total obstacles in its neighbourhood. Thirdly visibility has been defined using isovists lines [10] where mean sensor field of view is maximized for entire path. Multi-objective optimization study of path length and each of the safety measure is done on case studies. It is also shown how a generalized gene representation using B-splines can overcome the drawback of path monotonicity along one axis assumed in most studies [4] and solve complex point to point path planning problems.

II. REPRESENTATION

For successful implementation of any multi-objective optimization algorithm the foremost requirement is meaningful representation of path in terms of variables. Different methodologies [4], [5], [6] for such representation have been used in literature for single and multi objective study. [4] uses fixed length binary genes for monotonous paths on a grid. In there representation each sub-part of binary gene represents direction and displacement values of transition from one column to next in a grid. In such methods path is an aggregation of straight line segments leading to irregular paths which an autonomous vehicle may not be able to traverse. In many mobile autonomous vehicle's heading is proportional to the first derivative of the path while the second derivative is proportional to steering angle [7]. Discontinuities may cause the vehicle to stop and adjust its steering hence considering the control and dynamics constraints a path with continuous derivatives is essential. Different smooth path representation methods for path planning are used in [7], [8] and [9].

One of the possible solutions for smooth curve representation of path is polynomial curve fitting on way-points between start and destination through which path must pass. The coordinates of the way-points are taken as variables in optimization. Such a method would fail to represent paths with large number of wiggles which a polynomial of large degree may also fail in representing.

978-1-4577-2138-0/11/$26.00 © 2011 IEEE

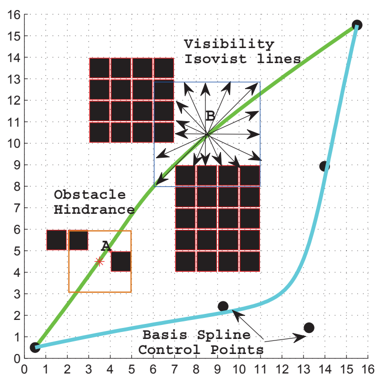

Fig. 1. Basis Splines and its control points. Hindrance detection at point A and isovist lines for local visibility at point B

Hence this paper details the implementation of piecewise polynomial splines (PP-splines) and basis splines (B-splines) as a tool for the multi-objective study.

A. Splines

Splines are piecewise polynomial functions defined through a finite set of control points. We represent the path by splines between different breakpoints or knots from start to destination.

Initially piecewise polynomial splines are used where a set of $2n$ real numbers represent the x and y coordinates of n way-points through which spline must pass. C^1 and C^2 continuity is maintained hence ensuring a smooth resultant path. Case study in IV-A uses PP-spline for multi-objective optimization. The disadvantage observed using this representation was that many individuals (paths) of the population overshoot outside the grid. The way-points do not have good control on the bounds of entire spline curve. Such paths are deemed invalid as explained in II-B.

To counter above problem B-splines are used which are a generalization of the Bezier curve. A B-spline curve of degree m with n control points consist of $(n - m)$ bezier curve segments. The shape of the spline is controlled by its control points. The benefit is that no matter how we vary the control points the curve always lies within the convex hull of the control polygon. The polygon formed by connecting the successive control points with lines is called the control polygon. In all our simulations we have used uniform B-splines of order 4. A uniform B-spline is a curve where the intervals between successive control points are equal. The B-spline is clamped at start and destination using multiplicity of knots while other knots are uniformly placed between source and destination.

A set of $2n$ real numbers represent the x and y coordinates of n control points of the B-spline. A sample B-spline with 3 variable control points is shown in Fig. 1

The number of control points n for B-splines and piecewise polynomial spline is very crucial, affecting computational time and result accuracy. It is observed that n depends on obstacle distribution in different problems and in test cases n is chosen by fixing computational cost and observing improvement in pareto front for different n values. Interestingly for a desired path if we keep increasing n then the control polygon eventually converges on the path itself.

For simplicity we represent the terrain profile by a $k \times k$ grid and place obstacles in grid cells but the methodology can be applied to obstacles of any shape.

B. Path characteristics

We define a valid path with following characteristics

- It must begin at the start point
- It must reach the destination point
- It must not cross the grid outer boundaries
- It must not pass through an obstacle containing cell

To ensure that the resultant path to be a valid following measures are taken

- The start point is taken as the first spline node and destination point as the last spline node. To clamp B-splines knot multiplicity is used.
- Overshooting outside grid boundaries leads to penalty on all objectives proportional to overshoot. Hence genes gradually decrease overshoot and become feasible.
- Obstacle cells are given high potential. If path crosses a obstacle cell all objectives in multi-objective study are penalized proportional to number of obstacles crossed.

Obstacle crossing is intentionally not kept as a constraint since doing so rejects many good genes in initial generations. Penalization scheme proportional to number of obstacles crossed has been found to give significantly better results.

1) Path monotonicity: Path planning applications using GA usually take assumption of path monotonicity along one axis. It helps in fixed length gene representation [3], [4]. After this assumption, though the representation is not general but it is more efficient in handling most sparse obstacle grids. Such problems are more common in real outdoor path planning scenarios.

In our study we have assumed path monotonicity for initial test cases to improve optimization convergence time. This significantly reduces the search space hence speeding the GA in path search. To implement monotonicity while decoding a gene the the control points are taken in ascending order of there x coordinates (for x monotonicity).

A generalized representation of path by a gene is not possible in schemes where each grid cell through which path passes is represented in the gene due to absence of any knowledge about constraints on path length. For such a problem path may swing back and forth before reaching destination making it difficult to use fixed gene length. Using B-splines the same method is further extended to non-monotonous paths in IV-E. For this just like previous case we take $2n$ real numbers representing the spline control points but there sequence is also accounted. The splines are created using successive nodes irrespective of there coordinate values on the grid. Hence the resultant path is not constrained to be monotonic along any direction and it can come back and forth. Such implementation is possible for PP-splines also where way-points are variables and we evaluate each spline part for consecutive node points while simultaneously satisfying end condition for smoothness. The problem arises when the end conditions of curvature cannot be satisfied leading to discontinuities. Hence a more preferred solution is to use B-splines for non-monotonic cases. They ensure path smoothness and can easily cater to local shape behaviour by varying control points.

III. Methodology

We have used MATLAB implementation of NSGA-II algorithm [11] with control points as variables. The method finds a trade-off front for multiple objective functions using genetic algorithm. The initial population is generated randomly within the variable bounds specified. The next generation of the population is computed using the non-dominated rank and a distance measure of the individuals in the current generation. The method uses controlled elitist genetic algorithm which favours individuals with better fitness value as well as increase the diversity of the population by measure of crowding distance. Binary Tournament selection, uniform mutation and crossover are used as parameter settings. It is observed that results are sensitive to mutation. A balance between exploitation and exploration is maintained by suitably choosing mutation rate. Mutation rate between 0.03 to 0.08 for 5 to 8 control points has been found to give good results in test cases.

A. Path length

In case of PP-splines the polynomial equation of spline parts between various nodes is known. Line integral of each spline part to find part length is done, summation of which gives the exact path length. Such an integration method for B-splines and PP-splines is computationally expensive hence path length is calculated using numerical approximation methods with straight line segment approximation on small intervals.

B. Path potential

Path safety on entire obstacle grid is defined in terms of potential field value. The grid is divided into $k \times k$ cells and each cell is allotted a potential value. The potential value is proportional to difficulty faced in passing through a particular cell. Obstacle containing cells are given high potentials. It is unsafe for a robot to just brush through an obstacle containing cell hence all neighbouring cells of an obstacle containing cell are also allotted a potential value. In this way the entire grid is represented in the form of a potential field. The potential field can be customized specific to actual problem environment and requirements of user. Potential field of surroundings due to obstacles can be given by different models using euclidean distance, Gaussian distribution or linearly decreasing potential value. The difficulty level of cells far away from a particular obstacle containing cell are not affected by it hence they have zero potential due to it. The total difficulty of a path is summation of potential field values the path traverses.

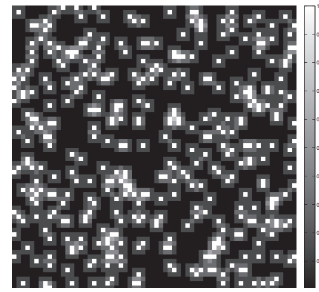

Fig. 2. Potential field for obstacle grid of IV-B

The initial grid cells are large in size hence grid side is divided into $\eta \times k$ cells and spline is dicretized on it. Potential per unit area for each cell is multiplied by area traversed by the path in that cell. Summation of these potential values over the grid gives net difficulty. To reduce computational time we have used matrix representation methods for obstacle field, potential fields and resultant path. Simple matrix operations in MATLAB help in quick evaluation of objectives.

In our simulations we have given each obstacle cell potential value 1 and all its empty neighbouring cells potential value of 0.15. Hence an empty cell with 3 obstacle containing cells in its immediate neighbourhood will have total potential of 0.45. Potential value of only immediate neighbours are affected by a particular obstacle. The potential field corresponding to obstacles in case study in IV-B is shown in Fig. 2. The potential values are shown in colour bar alongside.

C. Obstacle hindrance

The potential field approach is a flexible method to exactly define each cells potential value and how an obstacle is affecting the difficulty level in its surroundings. The problem arises when user input is not available to robot or a user does not have any knowledge of environment except presence or absence of obstacles. Hence we define another simple safety objective to minimize total surrounding obstacles for entire path. Suppose single cell neighbourhood is considered then at each path cell obstacles present in its 8 surrounding cells are called hindrance obstacles since they clutter the space and cause difficulty. Number of hindrance obstacles gives local hindrance value. In Fig. 1 local obstacle hindrance value at point A is 2. To calculate path hindrance all such obstacles in neighbourhood of the entire path are counted with each obstacle cell counted only once. Hence total hindrance value present is minimized as an objective. This ensures that robot will move at safe distance from obstacles as well as avoid narrow passages and cluttered environments. In IV-C path length and obstacle hindrance are conflicting objectives hence a trade off front is obtained giving the robot choice among possible optimal paths.

D. Visibility

For application areas which use mobile robots with on board sensors, visibility becomes an important criteria for path planning. Most robots use either camera vision or proximity sensors like laser scanner or infra-red sensors for static and dynamic obstacle detection. Robot visibility at a point can be viewed as the area visible to its sensors. If visibility is high it is more probable that it will detect danger thereby enhancing safety as compared to case where most sensor range is blocked by obstacles. Hence visibility is a safety related measure that allows for a better risk perception and avoidance, such that maximizing visibility can potentially minimize risks.

Visibility at a point is defined by using set of lines of sight referred to as isovists lines which can be obtained by casting light rays, uniformly in all directions from a cell [10]. These lines intersect with grid boundary or surrounding obstacles. The length of isovist lines are summed up and divided by the total number of rays to give the local average visibility value. Mean visibility for entire path is calculated by summing all local visibility values from path cells and dividing by the number of cells traversed. Maximizing visibility for entire path will avoid congested surroundings hence enhancing safety.

Since proximity sensors have limited range so the maximum range of isovist lines can be given accordingly. Isovist lines for local visibility at point B is shown in Fig. 1 where range is restricted to 2 cell neighbourhood. Case study in IV-D shows trade-off front solutions for bi-objective optimization between path length and visibility.

IV. Results

A. Path length and potential using PP-spline

This case is similar to Benchmark 2 in [3] with 88 obstacle cells on a 16×16 obstacle grid. The potential field has been defined using rules mentioned in III-B. We have represented the piecewise cubic spline by 6 control points with 2 control points being fixed at source and destination. Hence the GA uses 8 real number variables for x and y coordinates of 4 variable points. Grid boundaries are defined as variable bounds.

The trade off front obtained with a population size of 60 and 200 generations is shown in Fig. 4, where each point represents a path. In Fig. 3 representative paths marked on trade-off front as A, B and C are shown. A is the shortest path while C is safest path obtained. The control points are also marked on the paths. The trade-off between the two objectives in obtained paths is clear from Fig. 3. Due to availability of space, a smooth trade-off between the two objectives is possible for this problem. Solutions having small distance are too close to the obstacles and safe solutions keep away from obstacles.

B. Path length and potential using B-spline

In this test case B-splines have been introduced to solve a complex scattered obstacle problem with 388 obstacles on a 64×64 grid. The objectives considered are path length and

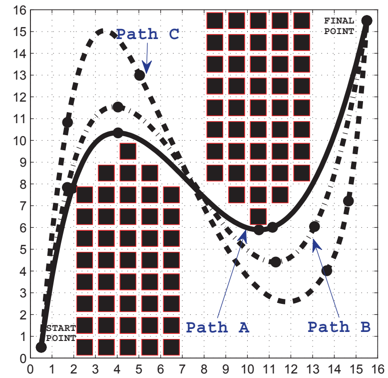

Fig. 3. Case study IV-A Obstacles on a 16×16 grid and few solution paths obtained on the pareto front

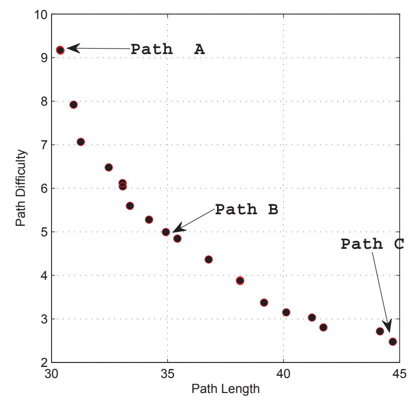

Fig. 4. Case study IV-A Trade off front for path length and potential

path potential. The corresponding potential field has been shown in Fig. 2. Uniform B-Spline with 8 control points is taken. The trade off front and corresponding paths obtained with a population size of 100 and 500 generations are shown in Fig. 5 and 6. It is noted that in this case solution paths look similar. This is because each obstacle affects potential up to 1 cell neighbourhood only. Due to large grid size differences between paths are not noticeable on this scale. It is interesting to note that all obtained trade-off solutions are members of two different principal paths – one for small distance and other for a better safety. There are small trade-offs between paths within each class. When search space is tightly constrained this is an usual phenomenon with multi-objective optimization. The front shown in Fig. 6 is also fragmented. The intermediate part of the front makes a small trade-off here and there between the two main principal paths.

C. Path length and hindrance using B-spline

This case considers a 32×32 grid with 245 obstacle cells to study the safety measure of obstacle hindrance along with path length. One cell neighbourhood (Fig. 1) is

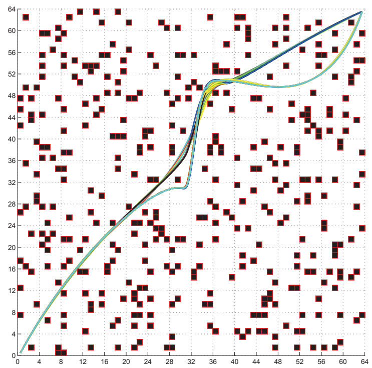

Fig. 5. Case study IV-B Obstacles on a 64×64 grid and solution paths obtained

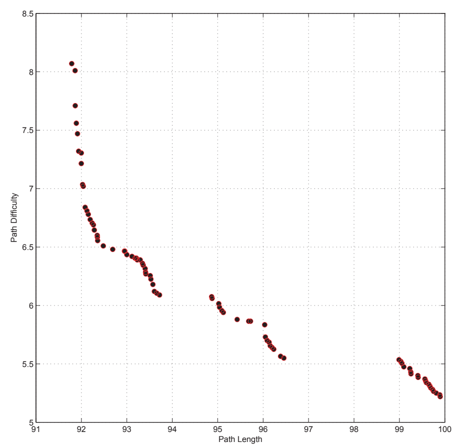

Fig. 6. Case study IV-B Trade-off front for path length and potential

considered for simulation and total hindrance obstacles are minimized along with path length. The conflict in path length and obstacle hindrance is evident from the resultant paths where to traverse the shortest path robot has to face many obstacles in its neighbourhood and pass through a small gap. B-Splines have been used for this bi-objective case with 5 control points and GA population size of 200 is used for 300 generations. Resultant paths obtained are shown in Fig. 7. Due to availability of intermediate spaces between the blocks of obstacles, principally different paths are obtained in this case.

D. Path length and visibility using B-spline

Analysis of visibility as a safety objective is done along with path length on a 32×32 grid. The aim is to get paths where robot sensors(vision or proximity) have large field of view as well as the resultant paths are short and smooth. 16 isovists lines from each path cell are taken and the maximum range of visibility sensor at a point is constrained inside a box of dimensions 5×5 cells as shown in Fig. 1. Using the dual principle instead of maximizing mean visibility and minimizing path length we have posed our problem as

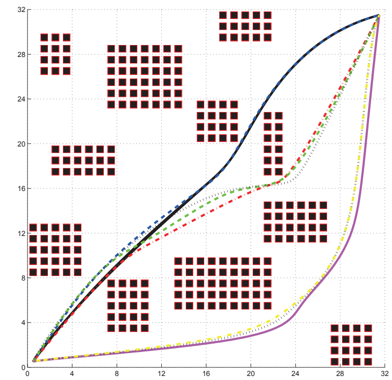

Fig. 7. Case study IV-C Solution paths obtained on minimizing path length and total hindrance

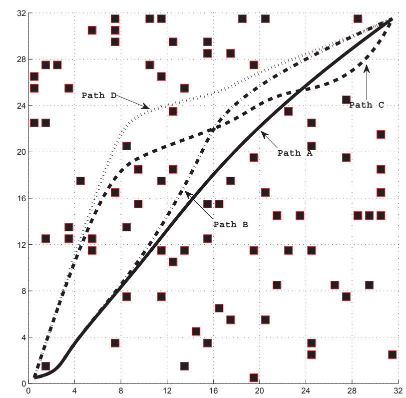

Fig. 8. Case study IV-D Solution paths from visibility and path length trade off front

minimization of path length and negative of mean visibility. 5 control points define the uniform B-spline and population size of 200 is used for 300 generations. The obtained trade off front is shown in Fig. 9 and four solution paths marked on it are shown in Fig. 8.

E. Non monotonic path using B-spline

Finally we have taken a difficult obstacle environment on a 64×64 grid with 887 obstacles which cannot be solved by one axis monotonic paths. To solve this B-spline is clamped at destination and start point and its control points are not taken in sequential order. Path length and safety potential are minimized using 8 control points. Population size and number of generations are fixed at 200 and 500. The corresponding solutions are shown in Fig. 10. From the solution we observe that in both x and y axes, non-monotonic paths are obtained. In this problem, there are not many principal paths possible and the obtained ones are the most viable ones. Slight variations in paths come to make a trade-off between multiple objectives.

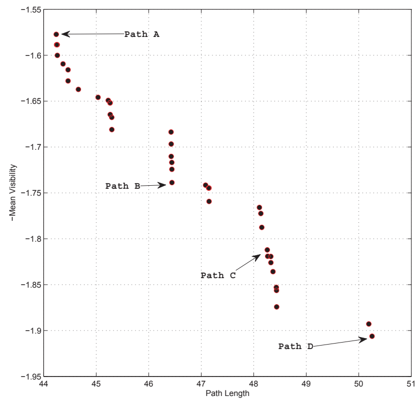

Fig. 9. Case study IV-D Trade off front for 2 objectives-path length and -visibility

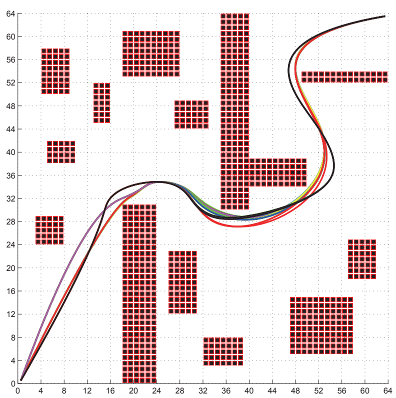

Fig. 10. Case study IV-E Non monotonic solution paths using B-splines on a 64×64 grid

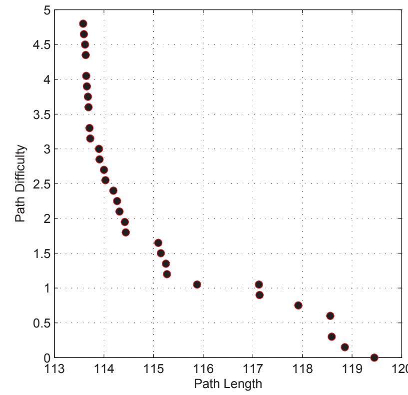

Fig. 11. Case study IV-E Trade-off front for path length and potential

V. Conclusion

We have used smooth path representation method using splines and defined different safety measures of potential, hindrance and visibility. Multi-objective analysis using NSGA-II for path length and each safety measure is done on test problems. The method gives the robot operator a set of choices for paths with varied length and safety measures from which final path can be selected. Besides finding trade-off paths, an analysis of the obtained paths (their numbers, principal shapes and their differences) can provide useful knowledge about how to plan a viable path under certain placement of obstacles. Once such knowledge is gained and learnt, they can be captured using some rule base or by some other means and used in dynamic path planning problems. This method can be extended using Non Uniform Rational Basis Splines (NURBS) and integrating it with such dynamic path planning methodologies.

References

[1] S.M. LaValle. *Planning algorithms*. Cambridge Univ Pr, 2006.
[2] Hwang Y, Ahuja N (1992) Gross motion planninga survey. ACM Computing Surveys (CSUR) 24(3):219–291
[3] Castillo O, Trujillo L (2005) Multiple objective optimization genetic algorithms for path planning in autonomous mobile robots. International journal of computers, systems and signals 6(1):48–63
[4] Sugihara K, Smith J (1999) Genetic algorithms for adaptive planning of path and trajectory of a mobile robot in 2d terrains. IEICE TRANSACTIONS on Information and Systems 82(1):309–317
[5] H. Jun and Z. Qingbao. Multi-objective mobile robot path planning based on improved genetic algorithm. In *2010 International Conference on Intelligent Computation Technology and Automation*, pages 752–756. IEEE, 2010.
[6] G. Dozier, S. McCullough, A. Homaifar, E. Tunstel, and L. Moore. Multiobjective evolutionary path planning via fuzzy tournament selection. In *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, pages 684–689. IEEE.
[7] J. Connors and G. Elkaim. Manipulating b-spline based paths for obstacle avoidance in autonomous ground vehicles. In *Proceedings of the ION National Technical Meeting*. Citeseer, 2007.
[8] Y.J. Kanayama and B.I. Hartman. Smooth local-path planning for autonomous vehicles1. *The International Journal of Robotics Research*, 16(3):263, 1997.
[9] H.C. Chang and J.S. Liu. High-quality path planning for autonomous mobile robots with η3-splines and parallel genetic algorithms. In *Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference on*, pages 1671–1677. IEEE, 2009.
[10] AR Soltani and T. Fernando. A fuzzy based multi-objective path planning of construction sites. *Automation in construction*, 13(6):717–734, 2004.
[11] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, 2002.