ENERGY EFFICIENT MOTION DESIGN AND TASK SCHEDULING FOR AN AUTONOMOUS VEHICLE

Xidias, Elias; Azariadis, Philip

Department of Product & Systems Design Engineering, University of the Aegean

ABSTRACT

This paper describes an approach for designing an energy efficient motion and task scheduling for an autonomous vehicle which is moving in complicated environments in industrial sector or in large warehouses. The vehicle is requested to serve a number of workstations while moving safely and efficiently in the environment. In the proposed approach, the overall problem is formulated as a constraint optimization problem by using the Bump-Surface concept. Then, a Pareto-based multi-objective optimization strategy is adopted, and a modified genetic algorithm is developed to determine the Pareto optimum solution. The efficiency of the developed method is investigated and discussed through simulated experiments.

Keywords: Energy-Efficiency, Mechatronics, Motion Design, Artificial intelligence, Optimisation

Contact:
Xidias, Elias
University of the Aegean
Department of Product and Systems Design Engineering
Greece
xidias@aegean.gr

Cite this article: Xidias, E., Azariadis, P. (2019) 'Energy Efficient Motion Design and Task Scheduling for an Autonomous Vehicle', in *Proceedings of the 22nd International Conference on Engineering Design (ICED19),* Delft, The Netherlands, 5-8 August 2019. DOI:10.1017/dsi.2019.292

https://doi.org/10.1017/dsi.2019.292 Published online by Cambridge University Press

1 INTRODUCTION

Nowadays, there is an emerging need for fully Autonomous Vehicles (AVs) which can perform complex tasks in complicated environments in industrial sector or in large warehouses. Such vehicles should be able to serve various workstations, while moving safely and efficiently in their environment. However, developing the appropriate algorithms for the AVs raises many critical, complex and combinatorial optimization problems.

Autonomous Vehicles usually consist of batteries, motors, motor drivers and controllers. Energy conservation can be achieved in several ways (Mei *et al.*, 2004):

- using energy-efficient motors,
- improving the power efficiency of motor drivers, and
- determining a velocity trajectory of the vehicle that reduces energy loss.

Direct current (DC) motors, are often used as actuators because they are small, cheap, reasonably efficient, and relatively simple to control, and can be directly supplied from the batteries. Furthermore, they are designed to be very efficient in their rated speed regions (Fabianski and Wicher, 2014). Because the motor speed is largely sensitive to torque variations, the energy dissipated in a DC motor of an AV is critically dependent on its velocity profile. Energy saving can be achieved by determining the optimal AV velocity trajectory (Shuaiby *et al.*, 2015).

Energy saving is one of the most important challenges for fully AVs powered by DC motors. It is a fundamental requirement to achieve long term deployment of AVs because one of their main parts is the on-board batteries which have limited lifetime. Thus, in order to extend the on-board batteries' lifetime, it is important to optimize the energy consumption of the vehicle (Mei *et al.*, 2005). Motion is a major source of energy consumption.

Sun and Reif (2005), consider the problem of computing an optimal path for a vehicle under the assumption that the friction coefficients are known across the terrain. In their work, the velocity and acceleration profiles are not optimized. In (Kim and Kim, 2007), the optimal velocity profile of an autonomous vehicle moving on a straight line is determined by assuming that the total travel time is fixed. Their solution does not incorporate any bound on maximum velocity of the AV.

In this paper, we examine the scenario where an AV is requested to serve a set of workstations in a planar indoor industrial environment. The AV starts from a depot location, passes through the requested workstations, exactly once, and returns to its original position. There are three objectives associated with this problem:

- Calculation of an optimum collision-free path.
- Calculation of an optimum task schedule.
- Calculation of energy efficient motion.

This problem is a combination of task scheduling and motion design. Hereafter, we will refer to this problem with the abbreviation EEMDTS (i.e., Energy-Efficient Motion Design and Task Scheduling).

To the best of our knowledge, the integration of these problems has been studied by very few researchers. In (Kiesel *et al.*, 2012), the integrated problem of AV scheduling and motion design is separated into three distinct stages: scheduling, building timetables, and routing. Using waypoints, where the AVs should pass through in specific time intervals, a timetable is calculated that specifies, for each waypoint, the time at which its assigned AV should arrive. The timetable is then passed to the router to find a safe path for each AV that achieves the given times. In (Xidias *et al.*, 2009), an autonomous vehicle is demanded to serve timely (providing delivery tasks) as many as possible workstations in a 2D industrial environment. The proposed methodology consists of two phases. First, the vehicle's environment is mapped onto a B-Spline surface embedded in 3D Euclidean space using a robust geometric model. Then, a modified genetic algorithm is applied for the determination of an optimum path that satisfies given mission constraints. However, this work does not take into account the corresponding kinematics constraints and the requirements of pickup and delivery of products at the workstations. Recently, Xidias (2018), proposed an approach for designing AV's near-optimum paths on weighted regions. An AV operating on an urban environment is requested to serve, in an optimum way, a set of customers. An A-star algorithm is implemented in order to construct a distance matrix between the depot and the customers and between the customer locations. Then, a Genetic Algorithm with special encoding is used to search for a near-optimum solution. However, this work does not take into account the AV's energy consumption.

https://doi.org/10.1017/dsi.2019.292 Published online by Cambridge University Press

In the present work, we propose an approach where:

- The AV can move in any direction in the indoor environment within the given constraints.
- Both motion design and scheduling are resolved simultaneously.
- The optimal path and the corresponding velocity profile are generated simultaneously.
- The AV's energy consumption is minimized.
- The overall problem is formulated as a multi-objective optimization problem (MOO) which is resolved using a Pareto-based approach. In this way, we are able to avoid dominating solutions that could lead to local optima and would require a tedious fine-tuning of weight and normalizing parameters of the objective functions (Chen and Ho, 2005).

The remainder of the paper is organized as follows: Section 2, provides a formal definition of the optimization problem under consideration. Section 3 analyses the motion planning and task scheduling. Section 4 presents the optimization algorithm for solving the formulated problem, while Section 5, reports and discusses the application of the proposed approach in simulated experiments. Finally, Section 6 summarizes the contribution of the paper and states some goals for future work.

2 PROBLEM FORMULATION

The EEMDTS problem is directly motivated by two questions which are arising in a modern warehouse or in an industrial environment where an AV is requested to serve several workstations. The questions are:

1. What's the optimum schedule for the vehicle to accomplish the requested tasks?
2. What's the optimum motion and velocity profiles for the vehicle that minimize the energy consumption?

In the following, subsection 2.1 defines the problem, subsection 2.2 describes the considered autonomous vehicles and related assumptions, and subsection 2.3 briefly presents the Bump-Surface (Azariadis and Aspragathos, 2005) concept that it is used to represent the floor environment and the solution space.

2.1 Problem definition

We consider an AV which is operating in a planar industrial environment cluttered with M workstations $\boldsymbol{WS} = \{WS_1, WS_2, \ldots, WS_{M-1}, WS_M\}$. A management system requests from the AV to serve a predefined number of workstations (Figure 1). According to this demand the following requirements should be met:

- The vehicle must serve a predefined subset of workstations $\boldsymbol{WS}_o \sqsubseteq \boldsymbol{WS}$ and each workstation $WS_i \in \boldsymbol{WS}_o$ should be served only once.
- The vehicle's motion starts from the depot and terminates at the depot.
- The locations of the depots and the workstations are fixed and known.
- The environment is cluttered with known obstacles with fixed geometry.
- The AV is moving in the environment with variable speed $u \in [0, u_{max}]$, where $u = 0$ holds at the depot and workstation locations, and $u = u_{max}$ is the maximum allowed speed determined according to safety regulations.
- The vehicle has an acceleration $a \in [a_{min}, a_{max}]$, where $a_{min} < 0$ is the maximum deceleration and $a_{max} > 0$ is the maximum acceleration.
- The energy loss (transforming loss P_l) during the transformation of electrical energy to mechanical energy is not considered.

Under these requirements, with the EEMDTS problem we seek to determine the AV path that satisfies the following energy motion design and task scheduling criteria and constraints:

1. The path should not intersect with the obstacles and should result to a safe AV motion.
2. The path should be energy efficient (minimum curvature, optimal velocity profile and minimum motion power).
3. The path always starts at a depot, passes through all the requested workstations, from each one exactly once, and returns at the same depot.

https://doi.org/10.1017/dsi.2019.292 Published online by Cambridge University Press

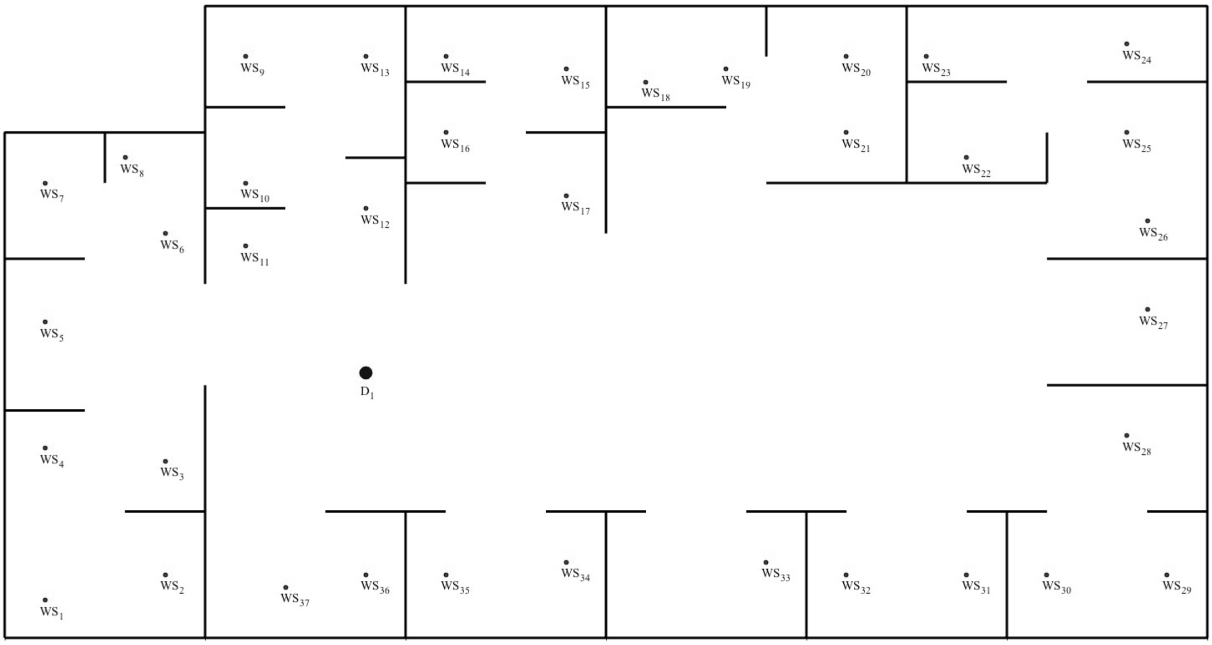


Figure 1. A 2D industrial environment cluttered with one depot (D_1) and a set of 37 workstations.

The aforementioned requirements form a set of conditions which are usually met in real world applications. Common examples include indoor industrial environments or warehouses, where AVs are requested to deliver products to a set of manned workstations.

2.2 The autonomous vehicle

In industrial applications, an AV is used to transport materials or accomplish specific tasks in many different industrial settings. They feature batteries or electric powered motors and computer technology that has been programmed to drive to and from specific points.

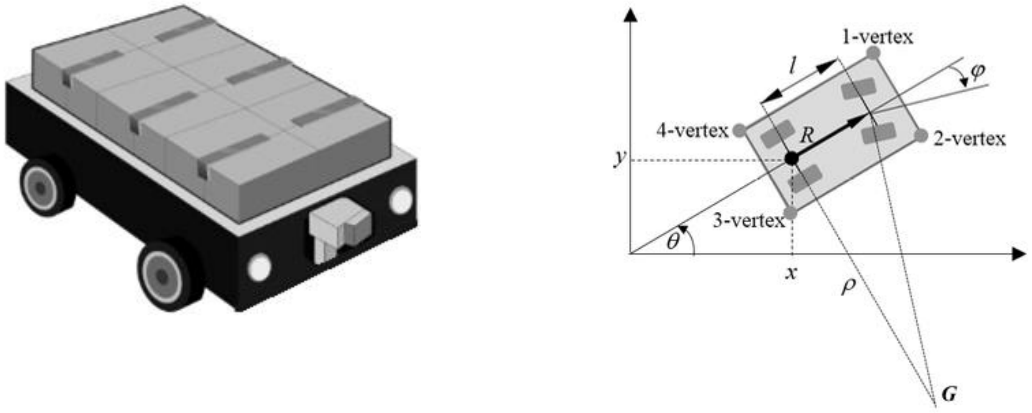
Throughout this work we consider that the AV has a car-like steering with forward, translational velocity provided by a DC motor (Figure 2). The AV can move "freely" in any direction in order to accomplish its task, with respect to EEMDTS criteria and constraints. For detailed information on this model the reader is referred to (LaValle, 2006).

The AV's configuration in the 2D environment is uniquely defined by $(x, y, \theta) \in \mathbb{R}^2 \times [0, 2\pi)$ where (x, y) are the Cartesian coordinates of the reference point R with respect to a fixed frame $\langle x, y \rangle$ and θ represents the orientation of the vehicle's chassis. φ denotes the steering angle, where $|\varphi| \leq \varphi_{max}$, l is the distance between the front and rear wheel axles and $\rho = {}^{1}\!/_{tan\varphi}$ is the radius of curvature at R. Note that ρ can be positive or negative according to the sign of φ. For example, if $\varphi < 0$ then $\rho < 0$ as well, which means "turn right".

The vehicle's wheels may slip when it is making a sharp turn at a high speed. The maximum speed u_{max} with which the vehicle can move along the vehicle's path is a function of the instantaneous ρ, the inertia of the vehicle and the frictional forces with the surface. We assume the maximum centrifugal force without slipping can be specified by a parameter F_{max}. Thus, the maximum safe translational speed without slipping will be (Tokekar *et al.*, 2014),

$$u(s)_{max} = \sqrt{\frac{F_{max}\,\rho(s)}{m}}, s \in [0,1] \tag{1}$$

where m is the mass of the vehicle.



(a) *(b)*

Figure 2. (a) An autonomous vehicle. (b) An overview of car-like model.

2.3 The workspace representation

Given a 2D industrial/warehouse environment, a normalized workspace $\boldsymbol{W}$ is constructed by linearly mapping the initial environment to $[0,1]^2$. The construction of the corresponding Bump-Surface (Azariadis and Aspragathos, 2005) is obtained by discretizing $\boldsymbol{W}$ into uniform subintervals along its x and y orthogonal directions, respectively, forming a grid of points $\boldsymbol{p}_{ij} = (u_{ij}, v_{ij}, w_{ij}) \in [0,1]^3$, $i, j \in [0, N_g - 1]$, where N_g is the (user-defined) grid size. The third coordinate w_{ij} of each $\boldsymbol{p}_{ij}$ is defined as follows:

$$w_{ij} = \begin{cases} (0,1], if\ \boldsymbol{p}_{ij}\ lies\ inside\ an\ obstacle \\ \qquad 0, otherwise \end{cases} \tag{2}$$

In this paper, we use a (2,2)-degree B-Spline surface with optimized knot vectors (Xidias and Azariadis, 2011) to represent the Bump-Surface $S : [0,1]^2 \to [0,1]^3$. Hence,

$$S(x,y) = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} N_i^2(x) N_j^2(y) \boldsymbol{p}_{ij} \tag{3}$$

where, $N_i^2(x)$ and $N_j^2(y)$ are the B-Spline basis functions (Piegl and Tiller, 1997). The constructed 3D surface S consists of 2D flat areas and 3D bumpy areas which correspond to environment's obstacles.

3 ENERGY EFFICIENCY IN MOTION DESIGN

As mentioned in Section 1, the energy consumption can be achieved in several ways, for example,

- using energy-efficient motors,
- improving the power efficiency of motor drivers, and
- finding better trajectories.

In this paper, we focus on the effect of motion planning to energy consumption. In order, to minimize energy consumption, we should design "optimal" paths for the vehicles. Optimality here is related to the amount of time spent to travel along the planned path and to the distance of the planned path. So, the goal is to generate optimal paths of minimum distance and travel time. A short-length path that contains several turns is not always energy efficient since the vehicle may consume more energy due to frequent decelerations and accelerations. In contrast, a path with longer length may require less energy if the vehicle does not have to accelerate and decelerate often.

We consider that, the midpoint of the rear wheels traces a path $\mathbf{R}(s) = (x(s), y(s))$ in $\boldsymbol{W}$ which is represented as a second degree NURBS curve (Piegl and Tiller, 1997),

$$\mathbf{R}(s) = \frac{\sum_{i=0}^{K-1} N_i^2(s) \omega_i \boldsymbol{p}_i}{\sum_{i=0}^{K-1} N_i^2(s) \omega_i}, s \in [0,1] \tag{4}$$

https://doi.org/10.1017/dsi.2019.292 Published online by Cambridge University Press

where, $N_i^2(s)$ is the B-Spline basis function, ω_i is the weight factor and $\boldsymbol{p}_i$ are the K control points defined as in the following:

- $\boldsymbol{p}_0 = \boldsymbol{p}_{K-1}$, denote the depot point.
- $\{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_{K-2}\} = \{\textit{the predefined subset of workstations}\} \cup$ $\{\textit{a number of intermediate points } g_j, j = 1,..,b\}$.

The number of intermediate points $g_j, j = 1,..,b$ is user-defined and depends on the complexity of the environment.

3.1 Computing a collision free path

A collision-free path that passes through the predefined subset of workstations should be searched in the flat areas of the Bump-surface. A path that "climbs" the bumps of the Bump-surface results in an invalid path in the initial 2D environment because it penetrates the obstacles. By construction, the arc length of $\mathbf{R}(s)$ approximates the length of its image $S(\mathbf{R}(s))$ on S, as long as $\mathbf{R}(s)$ does not penetrate the obstacles (Azariadis and Aspragathos, 2005). Therefore, it is reasonable to search for a flat path on S. This requirement can be written as:

$$E = Le^{\sum_{i=1}^{4} H_i}, s \in [0,1] \tag{5}$$

where, L is the arc length of $S(\mathbf{R}(s))$ and $H_i, i = 1, \ldots, 4$ is the flatness of the vehicle i –vertex (Figure 2) (Xidias and Azariadis, 2011). E takes a value in the interval $(L, +\infty)$ if the vehicle collides with obstacles and a value equal to L otherwise.

3.2 Defining a smooth path

In order to ensure a smooth path $\mathbf{R}(s)$, the corresponding curvature $C(s)$ should comply the AV kinematics constraints, i.e., $C(s) \le C_{max}$, where C_{max} is the maximum allowed curvature. We use the discrete curvature $C(s)$ definition according to (Kobbert, 1996):

$$C(s) = \left\| \mathbf{R}^{i-1} - 2\mathbf{R}^{i} + \mathbf{R}^{i+1} \right\|, i = 1, \ldots, N_p - 1 \tag{6}$$

where, the path $\mathbf{R}(s)$ is approximated by $N_p - 1$ sequential line segments.

3.3 Travel time computation

The AV's velocity $u(s)$ is constraint by the relation $0 \le u(s) \le u_{max}$. Velocity can never become negative and can be equal to zero only at the depot and at the workstation locations. The measure of the velocity $u(s)$ is defined by,

$$u(s) = \begin{cases} u_{max}, if\ p(s) = 0 \\ \sqrt{\dfrac{F_{max}\rho(s)}{m}}, if\ p(s) \neq 0 \end{cases} \tag{7}$$

where, u_{max} is the maximum allowed tangential velocity.

The infinitesimal time † t for the AV to move between two sequential path positions $\mathbf{R}^i$ and $\mathbf{R}^{i+1}$ is given by,

$$\Delta t^i = \frac{\left\| \mathbf{R}^{i+1} - \mathbf{R}^{i} \right\|}{avgu^i}, i = 1, \ldots, N_p - 1 \tag{8}$$

where, $avgu^i = (u^i + u^{i+1})/2$ is the average velocity of the vehicle between the points $\mathbf{R}^i$ and $\mathbf{R}^{i+1}$.

The travel time of the vehicle along the planned path $\mathbf{R}(s)$ is calculated by integrating (8) as a function of the parameter s and is given by,

$$t = \sum_{i=0}^{N_p-2} \frac{\left\| \mathbf{R}^{i+1} - \mathbf{R}^{i} \right\|}{avgu^i} \tag{9}$$

https://doi.org/10.1017/dsi.2019.292 Published online by Cambridge University Press

3.4 Computing the power consumption of the motors

The AV traverses along the path $\mathbf{R}(s)$ by using motors that consume energy in term of electrical power. The motors transform electrical energy into mechanical energy. The power consumption of the motors is the sum of the transforming loss (in the present work we set it equal to zero) and the mechanical output power. The motion power (P_m) can be modelled as the function of speed, acceleration and vehicle's mass as defined by the equation (Anuntachai *et al.*, 2014):

$$P_m(m,u(s),a(s)) = \sum_{i=0}^{N_p-1} P_l + m\sum_{i=0}^{N_p-1}(a^i + g\,\mu)u^i \tag{10}$$

where P_l is the transforming loss, g is the gravity constant, a^i is the instantaneous acceleration at point $\mathbf{R}^i$ and μ is the ground friction constant.

3.5 The overall problem

Taking the above analysis into consideration, the overall EEMDTS problem defined in Section 2 is formulated as a Pareto-based multi-objective optimization problem as in the following:

$$minimize\, E_{comp} = \{E,t,P_m\} \tag{11}$$

$$\text{subject to} \begin{cases} C(s) \leq C_{max} \\ 0 \leq u(s) \leq u_{max} \end{cases}$$

4 OPTIMIZATION METHODOLOGY

The EEMDTS problem can be characterized as a NP-complete multi-objective optimization problem. Due to the combinatorial explosion, the extraction of exact optimal solutions for NP-hard problems is computationally impracticable. Thus, considerable attention has been paid to combinatorial optimization based on metaheuristics, such as Genetic Algorithms, that seek approximate solutions in polynomial time instead of exact solutions which would be at intolerably high cost.

Genetic Algorithms (GAs) (Holland, 1995) are probabilistic search methods that employ search techniques inspired by Darwin's evolutionary theory based on the principles and mechanisms of natural selection and the survival of the fittest. GAs employ a random, yet directed, search for finding the globally optimal solution. They have the advantage over the gradient descent techniques that they do not require the derivative of the objective function and the search is not biased towards the locally optimal solution. In contrast to random sampling algorithms, they have the ability to direct the search towards relatively promising regions in the search. In addition, GAs have been proved to provide robust search even if the search space is NP-hard complete.

4.1 The chromosome syntax

The first step in applying a GA is the choice of an appropriate representation to encode the decision variables of the problem under consideration. In this work, a mixed integer/floating-point representation was selected for use. That is, chromosomes are vectors containing a set of successive integers followed by a set of successive real-valued numbers. Each integer number represents a workstation, and each real number represents a control point between two successive workstations or between a workstation and the AV's depot (Xidias and Azariadis, 2011).

4.2 The evaluation mechanism

A fitness assignment strategy based on Pareto-optimal solutions is proposed. The Pareto dominance relationship is formulated as follows: Let a possible solution Y and an objective function $\mathrm{F}_\mu(\mathrm{Y})$. Y_2 is a non-dominated solution and is said to dominate $\mathrm{Y}_1 : \forall\mu\text{: } \mathrm{F}_\mu(\mathrm{Y}_1) \geq \mathrm{F}_\mu(\mathrm{Y}_2)$ and $\exists\nu\text{: } \mathrm{F}_\nu(\mathrm{Y}_1) > \mathrm{F}_\nu(\mathrm{Y}_2)$. A feasible solution Y^* is said to be a Pareto-optimal solution if and only if no feasible solution Y exists that dominates Y^*.

In particular, in this paper a pure Pareto-ranking fitness assignment, called GPSIFF is used (Chen and Ho, 2005). In contrast to other strategies, the GPSIFF can assign discriminative fitness values not only to non-dominated individuals but also to dominated ones. The fitness function for a chromosome Y is formulated as follows: $fitness = p - q + c$, where p is the number of chromosomes that are dominated and q is the number of chromosomes that dominate the chromosome Y in the objective space, while c is a constant that is added to make fitness values positive. Figure 3, illustrates an example of fitness values of $6(c=6)$ participant individuals for a bicriteria optimization problem. For example, considering the individual A with a fitness value 7, in the rectangle formed by A, two individuals dominate A $(q=2)$ and three individuals are dominated by A $(p=3)$. Therefore, the fitness value of A is $3-2+6=7$.

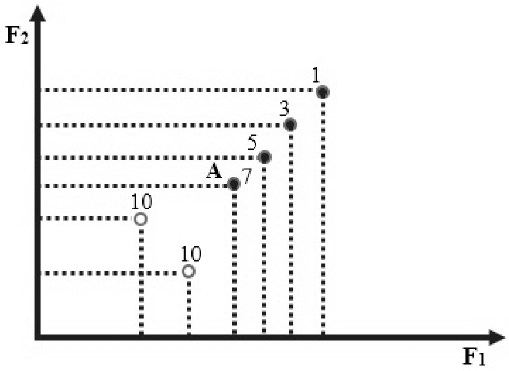


Figure 3. The calculated fitness function where the circles represent non-dominated solutions and the black dots are the dominated solutions.

4.3 Genetic operators

The following three genetic operators were selected for use with the proposed GA (Goldberg, 1989): **Reproduction**: Reproduction is a simple copy of an individual from the population of the current generation to the population of the next generation. In this work, the proportional selection strategy is adopted. According to this strategy, the chromosomes are selected to reproduce their structures in the next generation with a rate proportional to their fitness. **Crossover**: Crossover joins together parts of several individuals in order to produce new ones for the next generation. The individuals are randomly selected according to a user-defined probability (crossover rate). For the first part of the chromosome (integer values) the Order Crossover (OX) followed by a suitable repairing mechanism was selected for use, while for the second part of the chromosome (real values), the one-point crossover was adopted. **Mutation**: For the first part the inversion operator is used, while for the second part a boundary mutation is used. Inversion selects two positions along the string at random and then inverts the sub-section of the values between these two positions. Boundary mutation changes the value of a gene with a random number chosen from the permitted range of coordinates in $\boldsymbol{W}$.

4.4 Termination criterion

The proposed algorithm terminates either when the maximum number of generations is achieved or when the same best chromosome appears for a maximum number of generations.

5 EXPERIMENTAL RESULTS AND DISCUSSION

The performance of the proposed method is investigated through a number of simulation experiments for an AV which is moving in a 2D environment. Due to space limitations, we present only one example using the environment shown in Figure 1. The grid size is set to $N_g=100$ and we allow two intermediate points between every pair of workstations. The proposed GA was run using the following settings for the control parameters:

$\text{populations size} = 250$,

$\text{maximum number of generations} = 500$,

$\text{crossover rate} = 0.75$,

$\text{inversion rate} = 0.095$,

$\text{boundary mutation rate} = 0.004$.

The AV has maximum measure of velocity $u_{max}=1$ (velocity units) and maximum acceleration $a_{max}=1$ (acceleration units) and minimum deceleration $a_{min}=-1$ (acceleration units). The mass m of the AV is normalized to 1, while the ground friction constant is $\mu=0.7$ for the rubber wheel and concrete floor. Furthermore, we assume that the AV has the ability to make a turn at the workstation

https://doi.org/10.1017/dsi.2019.292 Published online by Cambridge University Press

and depot locations without violating the curvature constraint. It is worth noting that, the selection of the appropriate control settings was the result of extensive experimental efforts with various control schemes adopted following the indications of the literature.

In the present experiment a management system assigns to the AV the mission to serve a subset of 8 workstations $\boldsymbol{WS_o} = \{WS_{29}, WS_{36}, WS_{34}, WS_1, WS_{31}, WS_6, WS_{35}, WS_2\}$. All the workstations have the same requests of types of products. The presented approach concludes to two alternative solutions. In the first solution, the AV visits the workstations with the following order: $depot - WS_1 - WS_2 - WS_6 - WS_{29} - WS_{31} - WS_{34} - WS_{35} - WS_{36} - depot$. The corresponding solution vector is $E_{comp} = \{12.115, 16.11, 18.54\}$. In the second solution, the AV visits the workstations with the following order: $depot - WS_6 - WS_2 - WS_1 - WS_{29} - WS_{31} - WS_{35} - WS_{34} - WS_{36} - depot$. The corresponding solution vector is $E_{comp} = \{11.891, 15.98, 19.01\}$. Both solutions are illustrated in Figure 4. The first solution is represented by black solid line, while the second solution by black dashed line. The average velocity for the first solution is 0.834 (velocity units) while for the second solution is 0.812 (velocity units).

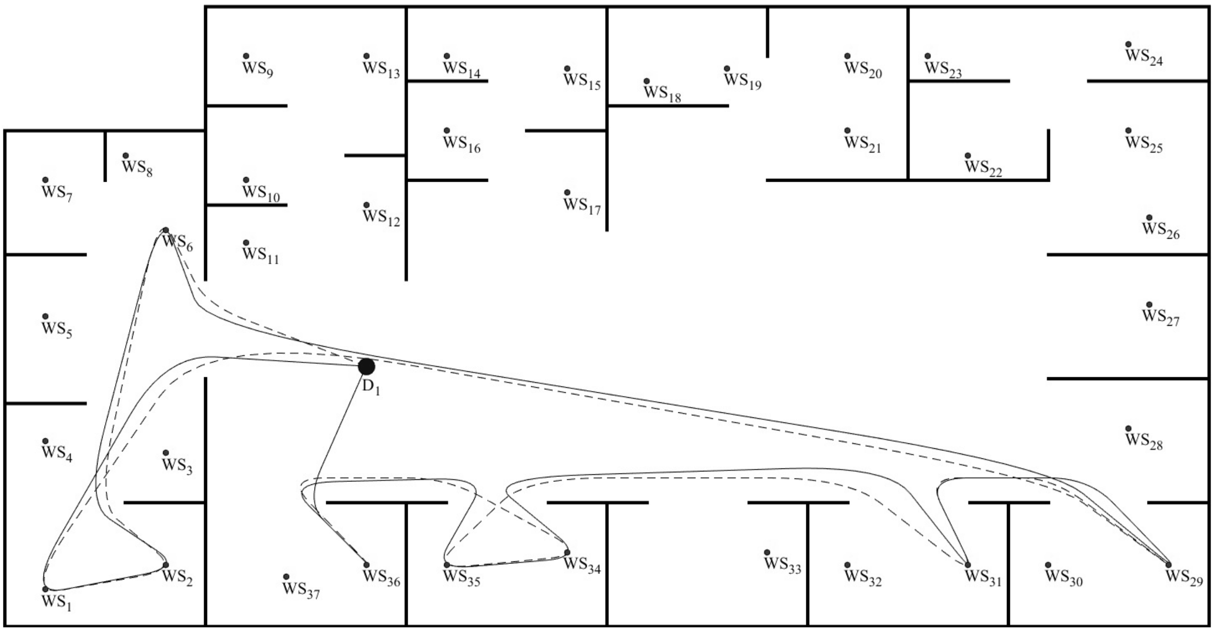


Figure 4. The proposed solutions.

In the above example we can see that, the proposed approach proposes two Pareto Optimal solutions, in other words solutions in which there exist no other solutions superior in all objectives. Comparing the two proposed solutions we can export the following conclusions: in both solutions the proposed paths are smooth and collision free, path's length in 1st solution is 1.84% greater from the 2nd solution, the travel time in 1st solution is 0.8% greater from the 2nd solution while the motion power in 1st solution is 2.53% smaller from the 2nd solution. Furthermore, comparing the proposed solutions with the common approach of combining the optimization criteria into a single objective function by using weights for linear combination of objective values we notice that, the derived solutions were very sensitive to small adjustments of the weight factors and produce suboptimal solution paths.

6 CONCLUSIONS

This paper presents a novel method for managing the motion of an autonomous vehicle used for logistics operations in indoors environments. The objective is to determine the energy efficient motion design and the task scheduling in order to serve a number of workstations. With the proposed method, the Task Scheduling and Energy Motion Planning problem is formulated and solved as a constrained optimization problem. A Pareto-based multi-objective optimization strategy is adopted, and a modified genetic algorithm is developed to determine the Pareto optimum solution. All our experiments show that, the proposed method is able to provide an optimum solution for an AV in complicated environments.

https://doi.org/10.1017/dsi.2019.292 Published online by Cambridge University Press

Future work will be concentrated on applying the proposed method in more complicated scenarios where a team of AVs are simultaneously operating in complex dynamic environments.

REFERENCES

Anuntachai A., Wongwirat O. and Thammano A., (2014), “An application of ant algorithm for searching energy-efficient route a mobile robot takes using energy as a weighting factor”, *Journal of Artificial Life and Robotics*, Vol. 19 No. 4, pp. 354–362. https://doi.org/10.1007/s10015-014-0175-8

Azariadis P.N. and Aspragathos N.A., (2005), “Obstacle representation by Bump-surfaces for optimal motion-planning”, *Robotics and Autonomous Systems*, Vol. 51 No. (2-3), pp. 129–150. https://doi.org/10.1016/j.robot.2004.11.001

Chen J.H. and Ho S.Y., (2005), “A novel approach to production planning of flexible manufacturing systems using an efficient multi-objective genetic algorithm”, *International Journal of Machine Tools & Manufacture*, Vol. 45, pp.949–957. https://doi.org/10.1016/j.ijmachtools.2004.10.010

Corréa A.I., Langevin A. and Rousseau L.M., (2007), “Scheduling and routing of automated guided vehicles: A hybrid approach”, *Computers & Operations Research*, Vol. 34 No. 6, pp. 1688–1707. https://doi.org/10.1016/j.cor.2005.07.004

Fabianski B. and Wicher B., (2014), “Dynamic model and analysis of distributed control system algorithms of three-wheel vehicle”, *In 19th International Conference on Methods and Models in Automation and Robotics (MMAR), 2-5 Sept*., https://doi.org/10.1109/mmar.2014.6957327

Goldberg D.E. (1989), “Genetic Algorithms in Search”, Optimization and Machine Learning. Addison Wesley Publishing Company. https://doi.org/10.5860/choice.27-0936

Kiesel S., Burns E., Wilt C. and Ruml W., (2012), “Integrating Vehicle Routing and Motion Planning”, *Proceedings of International Conference on Automated Planning and Scheduling*.

Kim, C. and Kim, B., (2007), “Minimum-energy translational trajectory generation for differential-driven wheeled mobile robots”, *Journal of Intelligent and Robotic Systems*, Vol. 49 No. 4, pp. 367–383. https://doi.org/10.1007/s10846-007-9142-0

Kobbert L. (1996), “A variational approach to subdivision”, *Computer Aided Geometric Design* Vol. 13, pp. 743–761.

LaValle M.S. (2006), *Planning Algorithms*. Cambridge University Press.

Mei Y., Lu Y.-H., Hu Y. C. and Lee C. G., (2005), “A Case Study of Mobile Robot’s Energy Consumption and Conservation Techniques”, *Proceedings of International Conference on Advanced Robotics*, pp. 492–497. https://doi.org/10.1109/icar.2005.1507454

Mei, Y., Lu, Y.-H., Hu, Y.C. and Lee, C.S.G. (2004), “Energy-efficient motion planning for mobile robots”, *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 5, pp. 4344–4349.

Piegl, L. and Tiller, W. (1997), *The NURBS Book*. Springer-Verlag Berlin Heidelberg. https://doi.org/10.1016/0010-4485(96)86819-9

Shuaiby, M., Akinori Y., Shigenori S. and Naoki U., (2015), “Simulation study on velocity distribution for energy saving of a redundant wheeled drive system”, *IEEE/SICE International Symposium on System Integration (SII)*, pp. 11–13 Dec. https://doi.org/10.1109/sii.2015.7404964

Sun, Z. and Reif, J., (2005), “On finding energy-minimizing paths on terrains”, *IEEE Transactions on Robotics*, Vol. 21 No. 1, pp. 102–114. https://doi.org/10.1109/tro.2004.837232

Tokekar, P., Nikhil K. and Volkan I., (2014), “Energy-optimal trajectory planning for car-like robots”, *Autonomous Robots*, Vol. 37, pp. 279–300. https://doi.org/10.1007/s10514-014-9390-3

Xidias E.K., (2018), “On Designing Near-Optimum Paths on Weighted Regions for an Intelligent Vehicle”, *International Journal of Intelligent Transportation Systems Research*. https://doi.org/10.1007/s13177-018-0159-5 https://doi.org/10.1007/s13177-018-0159-5

Xidias E.K. and Azariadis P.N., (2011), “Mission design for a group of autonomous guided vehicles”, *Robotics and Autonomous Systems*, Vol. 59 No. 1, pp. 34–43. https://doi.org/10.1016/j.robot.2010.10.003

Xidias E.K., Nearchou A.C. and Aspragathos N.A., (2009), “Vehicle scheduling in 2D shop floor environments. Industrial Robot”, Vol. 36 No. 2, pp.176–183. https://doi.org/10.1108/01439910910932630

https://doi.org/10.1017/dsi.2019.292 Published online by Cambridge University Press