Robotica (2016) volume 34, pp. 2257–2273. © Cambridge University Press 2015
doi:10.1017/S0263574714002872

Path Planning and scheduling for a fleet of autonomous vehicles

Elias Xidias†, Paraskevi Zacharia‡,* and Andreas Nearchou‡

†*Department of Product and Systems Design Engineering 84100, University of the Aegean, Hermoupolis, Syros, Greece*
‡*Department of Business Administration, University of Patras, 26 500, Rio, Patras, Greece*

(Accepted December 3, 2014. First published online: April 29, 2015)

SUMMARY
This paper presents a new solution approach for managing the motion of a fleet of autonomous vehicles (AVs) in indoor factory environments. AVs are requested to serve a number of workstations (WS) (following a specified desired production plan for materials requirements) while taking into account the safe movement (collisions avoidance) in the shop floor as well as time duration and energy resources. The proposed approach is based on the Bump-Surface concept to represent the 2D environment through a single mathematical entity. The solution of the combined problem of path planning and task scheduling is searched on a higher-dimension B-surface (in our case 3D) in such a way that its inverse image into the robot environment satisfies the given objectives and constraints. Then, a modified Genetic Algorithm (GA) is used to search for a near-optimum solution. The objective of the fleet coordination consists of determining the best feasible paths for the AVs so that all the WS are served at the lowest possible cost. The efficiency of the developed method is investigated and discussed through characteristic simulated experiments concerning a variety of operating environments.

KEYWORDS: Conflicts-free; Path planning; Motion planning; Logistics; Service robots.

1. Introduction

Automated guided vehicles (AGVs) are self-driven vehicles that follow markers or wires in the factory floor, or use vision or lasers without any accompanying operator. AGVs are widely used in material handling systems (MHSs), flexible manufacturing systems (FMS), and container handling applications. With the advance of technology, more sophisticated ("intelligent") machines, such as AVs are available, which considerably reduce machining and internal setup times. AVs are capable of fulfilling the transportation capabilities of a traditional forklift. They can sense their environment, navigating and scheduling their activities on their own.[2,17] Under these circumstances, the AVs fleet management system will accept in its input the description of the desired production plan for materials requirements on the various WS in the plant. Then, it will execute it (deciding and planning the best AVs routes through the WS) without any further intervention or help by the workers.

This paper considers a fleet of AVs moving in a 2D indoor industrial environment which is requested to serve a set of WS with various demands for delivery items. Each AV starts from its depot, passes through a number of WS and returns back to its depot. The objective is to determine the best vehicle safe-routes (collisions-free) so that all WS are served at the lowest possible cost. We will refer to this problem herein as the Vehicles Path Planning and Scheduling Problem (VPPSP).

The attainment of this objective necessitates the solution of two known combinatorial optimization problems: (a) the motion planning problem (MP) in dynamic environments, and (b) the vehicle routing and scheduling problem (VRS). Both of them are known to be intractable in the strong

* Corresponding author. E-mail: zacharia@mech.upatras.gr

https://doi.org/10.1017/S0263574714002872 Published online by Cambridge University Press

sense. MP and task scheduling issues are often studied separately. To the best of our knowledge, only few researchers have studied the integration of these problems; see for example.[5,16–18] In ref. [17], an autonomous robot is assigned to serve timely (providing delivery tasks) as many as possible WS in a 2D industrial environment. The proposed methodology consists of two phases: First, the vehicle's environment is mapped onto a 2D B-Spline surface embedded in 3D Euclidean space using a robust geometric model. Then, a modified GA is applied on the generated surface to search for an optimum legal path satisfying the requirements of the vehicle's mission. However, this work does not take into account the corresponding kinematics constraints of the robot and the requirements of pickup and delivery products at the stations. In ref. [5], a methodology is presented for modeling and controlling a flexible MHS, composed of autonomous robots, suitable for FMS. The autonomous robots incorporate artificial intelligence and mobile robotics techniques in order to calculate their paths. MHS makes use of a decentralized navigation control and a distributed Petri net in order to achieve higher flexibility and autonomy. However, the method is not globally optimal because for the task scheduling procedure is not taken into account for the generated paths. In ref. [16], a generic approach for the integration of VRS and MP for a group of autonomous robots is presented. The robots are requested to serve all the WS (ignoring the demands of the WS) in a 2D environment while taking into account kinematic constraints and environment's geometry during their motion. The objective is the simultaneous determination of time-optimum and collision-free paths for all the vehicles. However, this work ignores the requirements of pickup and/or delivery tasks. In ref. [18], a multi-resolution algorithm that seamlessly integrates the concepts of local navigation and global navigation based on the sensory information is presented. Here, the objective is to enable adaptive decision making and online replanning of vehicle paths. The proposed algorithm provides a complete coverage of the search area for cleanup of the oil spills and does not suffer from the problem of having local minima, which is commonly encountered in potential-field-based methods.

VRS is a well-known intractable logistics combinatorial problem. In its most simple form, the problem assumes characteristics identical to the traveling salesman problem with a single vehicle (having infinite capacity). The vehicle is assigned to visit a set of customers while the overall routing schedule satisfies some predefined time requirements. More general versions of the problem assume fleet of vehicles with capacity constraints, time windows for the delivery tasks, etc.[13] VRS is usually presented by an undirected graph and its solution is obtained by searching this graph for an optimum route satisfying the related constraints. All versions of VRS lead to an NP-hard optimization problem and therefore, the trend is to face these problems by using robust heuristics algorithms[14] or by using Petri-Net based approaches.[9,12]

MP in dynamic environments is a very challenging problem. Here, a robot should be able to plan its path and move toward to the desired location avoiding at the same time collisions with the environment and with other moving obstacles. Time optimal or near time-optimal approaches[7] for computing paths through state-time space have been developed; however, these algorithms are typically limited to low dimensional state spaces and/or require significant computation time. Thus, MP among moving and static obstacles is reduced to MP in a stationary environment. Time optimal or near time-optimal approaches for computing paths through state-time space have been developed, however these algorithms are typically limited to low dimensional state spaces and/or require significant computation time. MP approaches for a fleet of robots moving in known environments can be classified into centralized and decentralized approaches.[10] The trade-offs between these two categories lie on the efficiency and completeness of the corresponding methods.

1.1. Main contribution

In this paper, a new efficient approach for the solution of VPPSP is presented. The main characteristics of the developed approach are the following:

(i) An AV's tour is generated by taking into account: (a) the geometry of the environment, (b) the AV's kinematical constraints, (c) the characteristics of the logistics task operations (e.g., the WS' demands), (d) the AVs' energy constraints, and (e) the capacity of the robot.
(ii) The generated tours are smooth and collision-free without self-loops and sharp corners.

https://doi.org/10.1017/S0263574714002872 Published online by Cambridge University Press

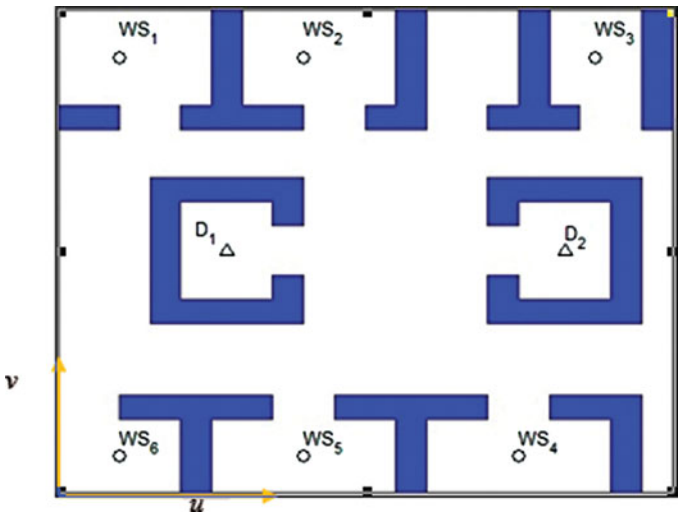


Fig. 1. A 2D industrial environment.

(iii) The overall optimization problem is formulated as a continuous system which is solved using a modified GA[4] with special encoding.

In contrast with the papers,[16,17] in this paper, we propose an approach for the generation of a management system. This system accepts as input the desired production plan for materials requirements on the various WS in the plant (e.g., the demands of the WS for the delivery quantities) and executes it (deciding and planning the near-optimum AVs paths through the WS) without any further intervention or help by the workers.

The remainder of the paper is organized as follows: Section 2 provides a formal definition of the optimization problem under consideration. Section 3 presents and analyzes motion planning and scheduling on the B-Surface. Section 4 presents a modified GA for solving the problem and Section 5 reports and discusses the efficiency of the proposed approach through simulated experiments. Finally, Section 6 summarizes the contribution of the paper and states some ideas for future work.

2. Formulation and Fundamentals

This section formally describes the VPPSP of a fleet of AVs which are requested to serve a set of WS in a 2D indoor industrial dynamic environment. Each AV is an autonomous car-like vehicle (rectangular shape body) moving with variable velocity. A table with all the notations used through the paper is given in the Appendix of the paper.

2.1. The VPPSP

A fleet of M AVs denoted by $\widehat{AV} = \{AV_1, ..., AV_M\}$ moves in a 2D normalized environment called W. W is assumed cluttered with static obstacles. Let $\widehat{D} = \{D_1, ..., D_G\}$ with $G \leq M$ be the set of the depots. More than one AV corresponds to each depot. L*et al*so $\widehat{WS} = \{WS_1, ..., WS_N\}$ (with $M \subseteq N$) be the set of WS in W, as illustrated in Fig. 1. Furthermore, the following assumptions and definitions are considered:

1. The obstacles have fixed and known geometry.
2. The only moving obstacles are the AVs.
3. The locations of both depots and WS are fixed and known.
4. Each AV is represented as a rectangular-shaped body with two rear wheels, two directional front wheels and a limited steering angle, for details see ref. [16].
5. Each AV has a capacity (>0) $CA_j > 0$ and moves only forward with a variable velocity $v_j(s) \leq v^{\max},\ j = 1, ..., M$.
6. Each WS is associated with a delivery quantity: $d_i \geq 0,\ i = 1, ..., N$.

https://doi.org/10.1017/S0263574714002872 Published online by Cambridge University Press

7. At each WS the AV is demanded to perform a delivery task.

Under these assumptions, VPPSP seeks to determine AVs' tours in W satisfying the following motion planning and scheduling criteria and constraints:

(a) Each tour should not intersect the obstacles and should result to a collision-free AV motion (*motion constraint*).
(b) Each tour should be time-optimum and smooth without sharp corners and self-loops (*energy constraint*).
(c) Each tour always starts at a depot and passes through as many as possible WS from each one exactly once and returns at the depot (*energy constraint*).
(d) A tour should have a lower-bounded turning radius (*kinematical constraint*).
(e) The total load along a generated tour should not exceed the j-AV's capacity $CA_j (j = 1, \ldots, M)$ (*capacity constraint*).

Note: *The aforementioned requirements form a set of conditions which are usually met in real-world industrial applications. For example, in an indoor industrial environment such as an automotive industry a fleet of AVs is requested to deliver products to a set of stations (assembly stations) where workers can pick them and finish the job. Throughout the day the workers stay in their stations while a continuous stream of robots deliver products to the stations enhancing the manufacturing line.*

2.2. The bump-surface concept

Given a 2D normalized workspace W, the construction of the Bump-Surface[1] corresponding to W is obtained by a straightforward extension of the Z-value algorithm.[1] Briefly, this algorithm considers that W is discretized into uniform subintervals along its u and v orthogonal directions, respectively, forming a grid of points $p_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j}) \in [0, 1]^3$, $i, j \in [0, N_g - 1]$, where N_g is the (used-defined) grid size. The third coordinate $(z_{i,j})$ of each $p_{i,j}$ takes a value in the interval $(0, 1]$, if the corresponding grid point lies inside an obstacle or the value 0 otherwise. In this paper, a $(2, 2)$-degree B-Spline surface with refined knot vectors is used[16] to represent the Bump-Surface $S : [0, 1]^2 \rightarrow [0, 1]^3$, which is given by

$$\mathbf{S}(u, v) = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} N_i^2(u) N_j^2(v) \mathbf{p}_{i,j}, (u, v) \in W \tag{1}$$

where, $N_i^2(u)$ and $N_j^2(v)$ are the B-Spline base functions.[11] The 3D surface S consists of 2D flat areas and 3D bumpy areas corresponding to the environment's obstacles. A visual example is depicted in Fig. 2.

3. Motion Planning and Scheduling on the Bump-Surface

Consider that the midpoint of the rear wheels of a j-AV traces a tour $\mathbf{R}_j(s) = (u_j(s), v_j(s)),\ j = 1, ..., M$ in W, which is represented as a two-degree NURBS curve,[11]

$$\mathbf{R}_j(s) = \frac{\sum_{i=0}^{K^j-1} N_i^2(s) \omega_j^i \mathbf{p}_i^j}{\sum_{i=0}^{K^j-1} N_i^2(s) \omega_j^i},\ s \in [0, 1],\ j = 1, ..., M \tag{2}$$

where $p_i^j = \{\tilde{\mathbf{p}}_0^j, \tilde{\mathbf{p}}_1^j, ..., \tilde{\mathbf{p}}_{K^j-2}^j, \tilde{\mathbf{p}}_{K^j-1}^j\}$ are the K^j control points. It holds that:

- $\tilde{\mathbf{p}}_0^j = \tilde{\mathbf{p}}_{K^j-1}^j$ denotes the depot point for the j-AV.
- $\{\mathbf{p}_1^j, ..., \mathbf{p}_{K^j-2}^j\} = \begin{pmatrix} \{subset\ of\ work\ stations\ \widetilde{WS}^j\} \cup \\ \{intermediate\ points\ g_i^j,\ i = 1, ..., b^j\} \end{pmatrix}$ defined in the parametric space of S.
- $N_i^2(s)$ represents the base function.

https://doi.org/10.1017/S0263574714002872 Published online by Cambridge University Press

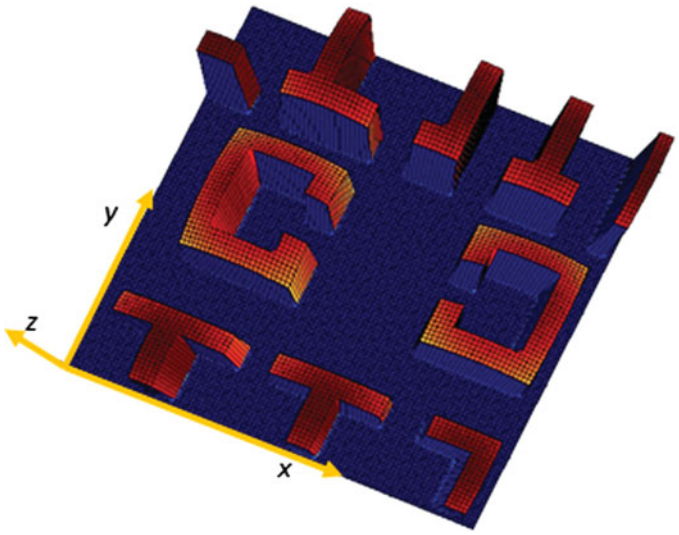


Fig. 2. The bump-surface corresponding to the 2D environment presented in Fig. 1.

- ω_j^i denotes the weight factors.
- $b^j = r(N_j + 1)$ denotes the total number of the intermediate control points $\mathbf{g}_l^j \in [0, 1]^2, l = 1, ..., b^j$ and N_j is the total number of $W\widehat{S}^j$.
- r is the number of intermediate control points g_l between two successive WS.
- In the given formulation, $W\widehat{S}^j \subset W\widehat{S}$ is the subset of WS to be served by the vehicle j-AV that follows the tour $\mathbf{R}_j(s)$.

Therefore, a solution to VPPSP consists of generating the $(K^j - 2)$ control points $\mathbf{p}_i^j$ such that the $\mathbf{R}_j(s)$, $j = 1, ..., M$ satisfies the motion planning and scheduling criteria and constraints.

3.1. Satisfying the motion constraint (constraint a)

A feasible j-tour that avoids the obstacles should be searched in the "flat" areas of the Bump-Surface. A j-tour that "climbs" the bumps of the Bump-Surface results in an invalid j-tour in the initial 2D environment that penetrates the obstacles. The arc length of the image of the $\mathbf{R}_j(s)$ onto S is given by:

$$L_j = \sum_{i=0}^{N_p-2} \left\| S(\mathbf{R}_j^{i+1}) - S(\mathbf{R}_j^i) \right\|, \; j = 1, ..., M \tag{3}$$

where $N_p - 1$ are equal sequential chords that approximate the curve $\mathbf{R}_j(s)$. In order to take into account the geometry of j-AV, a set of vertices $\mathbf{a}_j^k$, $k = 1, ..., n$, on the perimeter of j-AV is selected. The value n expresses the total number of vertices used to describe the geometry of the j-AV and are equally set on the perimeter of the j-AV. For all the examples analyzed in this study, we set $n = 4$.

Similarly with the midpoint $\mathbf{R}_j$, each point $\mathbf{a}_j^k$ follows a curve $\mathbf{a}_j^k = \mathbf{a}_j^k(s)$ in W. Then, following the results from,[16] we measure the "flatness" H_j^k of the image $\mathbf{S}_z(\mathbf{a}_j^k(s))$ of $\mathbf{a}_j^k(s)$ on S, i.e., $H_j^k = \int_0^1 \mathbf{S}_z(\mathbf{a}_j^k(s))ds$. Furthermore, in order to take into account the requirement that j-AV should avoid collisions with the other vehicles. Briefly, at each point $\mathbf{R}_j^i(s)$, $i = 1, ..., N_p$ of $\mathbf{R}_j(s)$, where N_p is the number of the selected points along $\mathbf{R}_j(s)$, it is assumed that a vehicle j-AV occupy a region Q_j^i in W, which is encircled with a circle centered at the center of j-AV and diameter d_j, where $d^j = \|\mathbf{a}_j^1 - \mathbf{a}_j^3\|$. If the distance d between the centers of two vehicles j-AV and j'-AV satisfies the $d \geq \frac{d_j + d_{j'}}{2}$, then no collision exists. Otherwise, collision detection requires searching for intersecting edges between the pairs of edges of the vehicles using the sweep line algorithm.

https://doi.org/10.1017/S0263574714002872 Published online by Cambridge University Press

Let $E_j = \frac{e^{(\sum_{k=1}^{4} H_j^k)} * L_j}{N_j}$ be a penalized length function corresponding to $\mathbf{S}(\mathbf{R}_j(s))$. E_j takes a value in the interval $(L_j, +\infty)$, if j-AV collides with obstacles and a value equal to L_j otherwise. Then, the requirement for a collision-free tour for the j-AV, which has minimum length and passes through the maximum number of WS, can be described as an optimization sub-problem with respect to p_i^j formulated as following:

$$\min E_j$$
$$\text{subject to} \bigcap_{j=1}^{M} Q_j^i = \emptyset, \ i = 1, ..., N_p. \tag{4}$$

Each j-AV should follow a path $\mathbf{R}_j(s)$ that eventually connects a number of work stations $\widehat{WS}^j$ with its depot location. Since every WS should be served by just one AV, then the following holds:

$$\widehat{WS}^1 \cup ... \cup \widehat{WS}^M = \widehat{WS} \text{ and } \widehat{WS}^1 \cap ... \cap \widehat{WS}^M = \emptyset. \tag{5}$$

3.2. Satisfying the energy constraint (constraint b)

The velocity $v_j(s)$ is constrained by the relation

$$0 < v_j(s) \leq v^{\max}. \tag{6}$$

The velocity $v_j(s)$ can never become negative and can be equal to zero only at the depot and the WS. Since $\mathbf{R}_j(s)$ is discrete, the measurement of the velocity v_j^i at every point $\mathbf{R}_j^i$ is defined by ref. [16]

$$v_j^i = \begin{cases} v^{\max}, if C_j^i = 0 \\ \min\left(v_j^{\max}, \sqrt{\frac{\tau_0}{C_j^i}}\right), if C_j^i \neq 0 \end{cases} \tag{7}$$

where, τ_0 is a constant which depends on the friction between the wheels and the ground, and the gravity constant g.

Let $\mathbf{R}_j^i$ and $\mathbf{R}_j^{i+1}$ be two sequential points of the discretized tour $\mathbf{R}_j(s)$. Furthermore, it is assumed that the j-AV moves from the point $\mathbf{R}_j^i$ to the point $\mathbf{R}_j^{i+1}$ in an infinitesimal time Δt_j^i, and ΔE_j^i is the corresponding displacement along $\mathbf{R}_j(s)$. The average velocity of j-AV during this period is represented by v_j^i.

The infinitesimal travel time Δt_j^i from point $\mathbf{R}_j^i$ to point $\mathbf{R}_j^{i+1}$ is given by

$$\Delta t_j^i = \frac{\Delta E_j^i}{v_j^i}. \tag{8}$$

Then, the time required for j-AV to travel along $\mathbf{R}_j(s)$ is calculated by

$$t_j = \sum_{i=0}^{N_p-2} \Delta t_j^i = \sum_{i=0}^{N_p-2} \frac{\Delta E_j^i}{v_j^i}. \tag{9}$$

Furthermore, the constraint **b** expresses the requirement for a smooth path $\mathbf{R}_j(s)$ (C^2 continuous) without sharp corners and loops. This requirement should be ideally followed by the optimum velocity profile with respect to the parameter s. These requirements are expressed by

$$E_j^{\text{int}} = \int_0^1 \left|\mathbf{R}_j'(s)\right|^2 dt + \int_0^1 \left|\mathbf{R}_j''(s)\right|^2 dt. \tag{10}$$

https://doi.org/10.1017/S0263574714002872 Published online by Cambridge University Press

E_j^{int} is considered as the *internal* energy of $\mathbf{R}_j(s)$ (also called as the strain energy). Minimizing E_j^{int} with respect to $\mathbf{p}_i^j$ leads to a j-tour which satisfies constraint (**b**).

3.3. Satisfying the kinematical constraint

In order to ensure that the curvature $C_j(s)$ along the $\mathbf{R}_j(s)$ never exceeds a maximum curvature $C_j^{\max}$ to avoid violating the kinematical constraints and force the moving j-AV velocity in the interval $(0, v_{\max}]$ the following condition should also hold:

$$C_j(s) \leq C_j^{\max} \ , s \in [0, 1] \, . \tag{11}$$

In this paper, $\mathbf{R}_j(s)$ is discretized by $N_p - 1$ sequential chords, and therefore the curvature C_j^i at the $\mathbf{R}_j^i = \mathbf{R}_j(s_i)$ point is approximated by the equation,[16]

$$C_j^i = \left\| R_j^{i-1} - 2R_j^i + R_j^{i+1} \right\| , \quad i = 1, ..., N_p - 1. \tag{12}$$

Thus, condition (11) is can be rewritten in a discrete manner as

$$C_j^i \leq C_j^{\max} \ , \quad i = 1, ..., N_p. \tag{13}$$

3.4. Satisfying the capacity constraint

Finally, in order to take into account the criterion (**d**), the following constraint is incorporated

$$\sum_{i=0}^{N_j} d_j^i \leq CA_j (j = 1, \ldots, M). \tag{14}$$

3.5. The overall problem

Following the above analysis, the overall optimization problem with respect to the control points $\mathbf{p}_i^j$ is written as

$$\min E_{\text{comp}}$$
$$\text{subject to} \begin{cases} C_j^i \leq C_j^{\max}, \quad i = 1, ..., N_p \\ \bigcap_{j=1}^{M} Q_j^i = \emptyset, \ \ i = 1, ..., N_p \\ \widehat{WS}^1 \cup ... \cup \widehat{WS}^M = \widehat{WS} \\ \widehat{WS}^1 \cap ... \cap \widehat{WS}^M = \emptyset \\ \sum_{i=0}^{N_j} d_j^i \leq CA_j \end{cases} , \ j = 1, ..., M \tag{15}$$

where the function E_{comp} is given by the vector

$$E_{\text{comp}} = \left\{ \sum_{j=0}^{M} E_j, \sum_{j=0}^{M} E_j^{\text{int}}, \sum_{j=0}^{M} t_j \right\} . \tag{16}$$

4. Solution Methodology

GAs have been successfully applied to optimization problems with large and complex search spaces due to their ability to reach a global near-optimal solution even if the search space contains multiple local minima.[6] Besides, GAs have extensively been used to address hard combinatorial optimization

https://doi.org/10.1017/S0263574714002872 Published online by Cambridge University Press

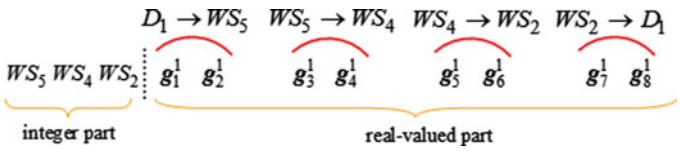


Fig. 3. The chromosome corresponding to the vehicle 1-AV.

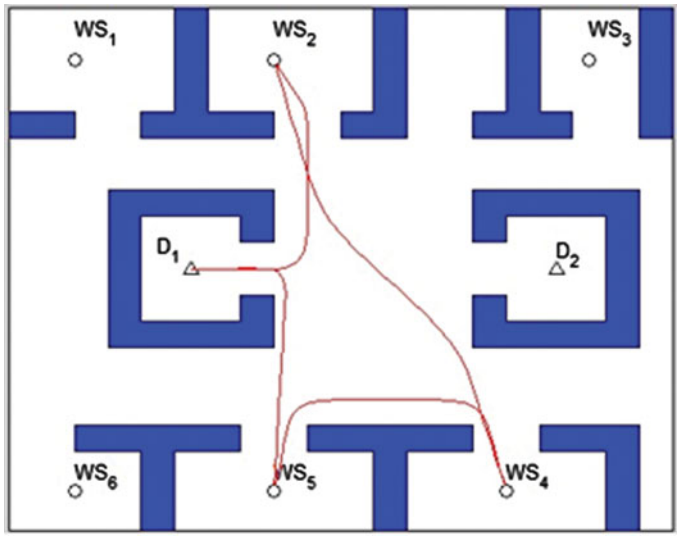


Fig. 4. A calculated path for vehicle 1-AV which corresponds to the chromosome of Fig. 3 The red closed curve shows the tour of the 1-AV.

problems including MP[15] as well as machine scheduling problems.[8] Thus, they have been successfully applied for the solution of optimization problems similar to that given in Eq (16). In this study, a modified GA for the solution of VPPSP is proposed. The main characteristics of the developed GA are analyzed and described in the following sub-sections.

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 is applied. That is, chromosomes are vectors containing a set of successive integers followed by a set of successive real-valued numbers. Each chromosome represents M possible paths for the set of AVs in the 2D environment. More specifically, each chromosome consists of $M + 2\sum_{j=1}^{M} b^j +$ cardinal $(W\widehat{S})$ genes, where $2\sum_{j=1}^{M} b^j$ is the overall number of the intermediate points used for the M generated paths. The integer part of the chromosome represents the order with which each j-AV visits the $W\widehat{S}^j$ WS. The real-valued part of the vector represents for each j-AV the b^j intermediate points $g_i^j \in W$. Figure 3 demonstrates the structure of the chromosome part corresponding to a random tour of vehicle 1-AV. The AV route corresponding to this chromosome is shown in Fig. 4. As one can see, the segments of the path between the three WS and the depot D_1 are determined by the two intermediate points $g_i^1 \in W$ $(i = 1, ..., 8)$ each pair of WS, where 8 is the total number of g_i^1.

4.2. The evaluation mechanism

In this paper, a fitness assignment strategy based on Pareto-optimal solutions is implemented. The Pareto dominance relationship is formulated as follows: let a possible solution Y and an objective function $F_\mu(Y)$. Y_2 is a non-dominated solution and is said to dominate Y_1: $\forall\mu : F_\mu(Y_1) \geq F_\mu(Y_2)$, and $\exists\nu$: $F_\nu(Y_1) > F_\nu(Y_2)$. 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 (GPSIFF) is used, inspired by the work of ref. [3]. In contrast to other strategies, the GPSIFF can assign discriminative fitness

https://doi.org/10.1017/S0263574714002872 Published online by Cambridge University Press

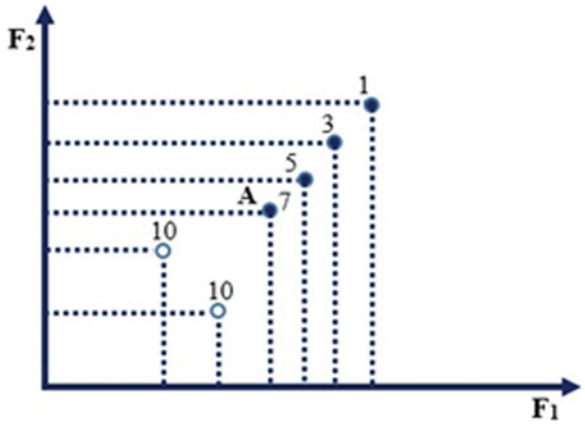


Fig. 5. The calculated fitness function where the circles represent non-dominated solutions and the black dots are the dominated solutions.

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 5 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 dominates A ($q = 2$) and three individuals is dominated by A ($p = 3$). Therefore, the fitness value of A is $3 - 2 + 6 = 7$.

4.3. Genetic operators

The following three genetic operators were selected for use with the proposed GA:

- ***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 (that with the integers) the order crossover (OX) followed by a suitable repairing mechanism was selected for use, while for the second part (real-valued part) of the chromosome, the one-point crossover was adopted.
- ***Mutation***: For the first part (with integers) the inversion operator is used, while for the second part a boundary mutation was 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 the B-surface.

4.4. Termination criterion

In most cases, a maximum number of iterations (generations) is defined in advance. However, it is difficult to determine beforehand the number of generations needed to find near-optimum solutions. Thus, an assessment of the quality level of the GA should be made online. 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. Simulation Results and Discussion

The performance of the proposed method is investigated through a number of simulation experiments for a set of AVs moving in 2D industrial environments. All simulations are implemented in Matlab and run on a Core 2 Duo 2.13 GHz PC. Four indicative examples are presented and discussed in

https://doi.org/10.1017/S0263574714002872 Published online by Cambridge University Press

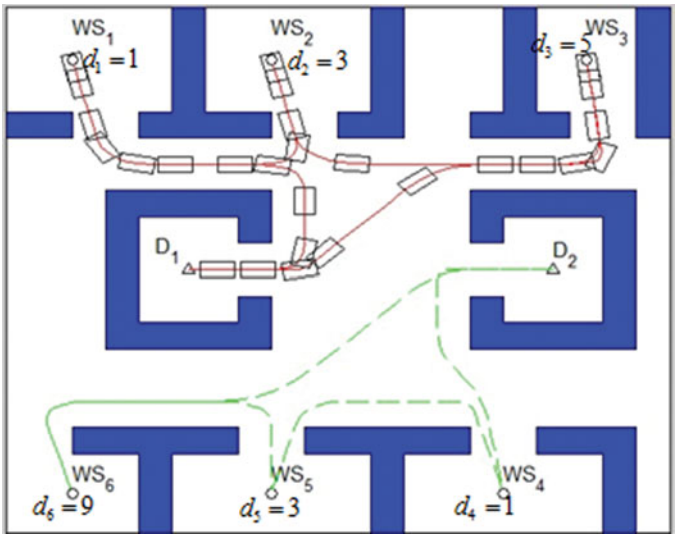


Fig. 6. The solution tours for the AVs.

this section. In all cases, the grid size is set to $N_g = 100$. The GA control parameter settings are: *population size* $= 250$, *maximum number of generations* $= 500$, *crossover rate* $= 0.75$, *inversion rate* $= 0.095$, and *boundary mutation rate* $= 0.004$. 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.

Furthermore, a (2, 2)-degree B-Spline surface is used to represent the Bump-Surface and $v_{\max} = 4$ (denoting the maximum allowed velocity for each AV). For the NURBS representation of $\mathbf{R}_j(s)$, the weight factors associated to WS are set equal to 3.2 and for the rest control points equal to 1. Under this way, it is possible to control the distance of an AV from a WS; (in our experiments this distance is of O(10^{-3}) order). Closer or looser proximity can be achieved by fine tuning these weights. This implies that the proposed method allows the assignment of different proximity weight-factors to the various WS. Finally, it is considered that every AV will spend the same amount of time in every WS and therefore this delay is not taken into account in the overall problem. Incorporating time frames in the current mission design strategy is considered as a goal for future research.

Test I: The first experiment corresponds to the environment shown in Fig. 1; a 2D indoor industrial environment containing several rooms with large halls and two depots. Two AVs are requested to deliver supplies to the WS. AVs have the same capacity quantity $CA = 15$. The total number of WS is $N = 6$. The number r of control points between each pair of WS is set to three ($r = 3$). Thus, the overall number of the unknown control points is 24. The tour solution (generated by the proposed GA) for the 1-AV is: $D_1 \rightarrow WS_1 \rightarrow WS_2 \rightarrow WS_3 \rightarrow D_1$ and the tour solution for the 2-AV is: $D_2 \rightarrow WS_6 \rightarrow WS_5 \rightarrow WS_4 \rightarrow D_2$. Both tours are displayed in Fig. 6. The construction of the Bump-Surface takes approximately 1.08 s of CPU time. The red closed curve shows the tour solution for the 1-AV and the green dashed closed curve shows the tour for the 2-AV. The velocity profile of the two AVs is depicted in Fig. 7.

Test II: The second experiment corresponds to the industrial environment shown in Fig. 8. Three AVs are requested to deliver in an optimum way supplies to the $N = 15$ WS. There are two depots used by the vehicles as follows: 1-AV and 3-AV use depot D_1 while 2-AV uses depot D_2. All the AVs have the same capacity quantity $CA = 15$. The number r of control points between each pair of WS is defined equal to $r = 3$. The solution tours generated by the proposed GA are (see Fig. 8) as follows:

- $D_1 \rightarrow WS_7 \rightarrow WS_1 \rightarrow WS_8 \rightarrow WS_2 \rightarrow WS_9 \rightarrow D_1$, (1-AV).
- $D_2 \rightarrow WS_{14} \rightarrow WS_4 \rightarrow WS_{15} \rightarrow WS_3 \rightarrow WS_{10} \rightarrow D_2$ (2-AV).
- $D_1 \rightarrow WS_{14} \rightarrow WS_4 \rightarrow WS_{15} \rightarrow WS_3 \rightarrow WS_{10} \rightarrow D_1$ (3-AV).

The Bump-Surface construction for this test case (shown in Fig. 10) takes nearly 1.22 s of CPU time. As far as the computational time is concerned, we observed that the GA spent for the solution of

https://doi.org/10.1017/S0263574714002872 Published online by Cambridge University Press

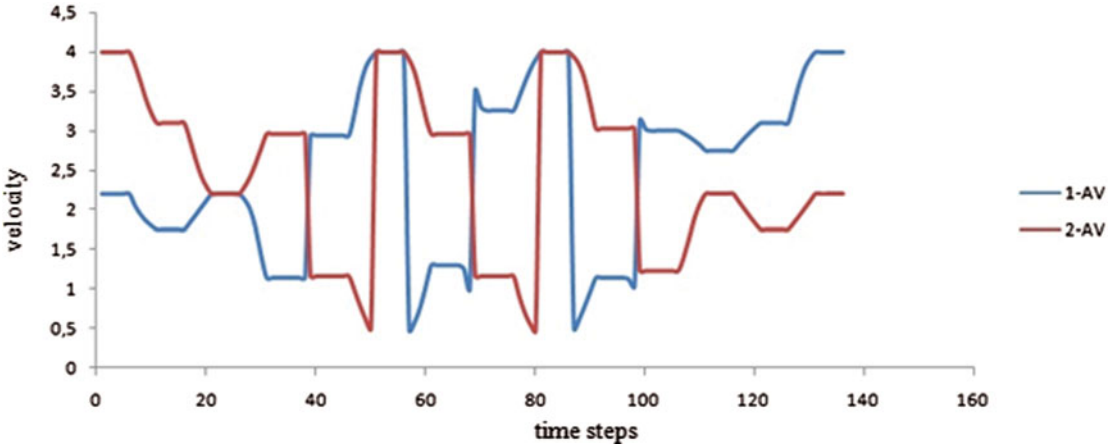


Fig. 7. The velocity profile of the two AVs.

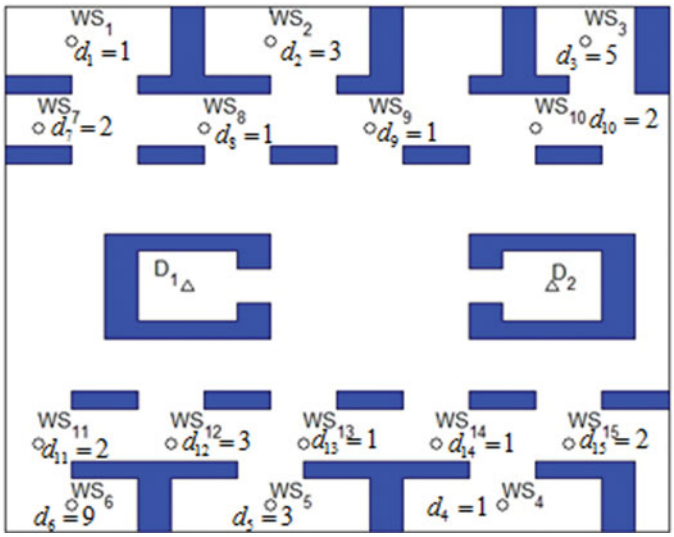


Fig. 8. A 2D indoor factory plant environment.

the second test problem approximately 9.8% more CPU time than that needed in the first test problem. The red closed curve (see Fig. 9) shows the solution tour for 1-AV, the green dashed closed curve shows the solution tour for 2-AV, and the blue one shows the solution tour for 3-AV. The velocity profiles of the three AVs are shown in Fig. 11.

Test III: A more complicated example is shown in Fig. 12. Here, we assume the environment used in test II, but more AVs and more depots have been added. Furthermore, AVs have now different capacity quantities. In particular, four AVs are requested to deliver supplies to 13 WS at minimum cost. Each AV starts from a different depot. The capacity of each vehicle has been defined as follows: $CA_1 = CA_2 = 8$, $CA_3 = 5$, $CA_4 = 6$. The number r of control points between each pair of WS is defined equal to $r = 4$. After running nearly 1.37 s, the proposed GA reached the solutions that are graphically shown in Fig. 13. The velocity profiles of the four AVs are illustrated in Fig. 14. As one can see from Fig. 13, the generated tour solutions for each vehicle are as follows:

- $D_1 \rightarrow WS_5 \rightarrow WS_1 \rightarrow D_1$ (1-AV, red closed curve in Fig. 13).
- $D_2 \rightarrow WS_{11} \rightarrow WS_{12} \rightarrow WS_3 \rightarrow WS_{13} \rightarrow D_2$ (2-AV, red dashed curve).
- $D_3 \rightarrow WS_9 \rightarrow WS_{10} \rightarrow WS_4 \rightarrow D_3$ (3-AV, green dashed curve).
- $D_4 \rightarrow WS_8 \rightarrow WS_7 \rightarrow WS_2 \rightarrow WS_6 \rightarrow D_4$ (4-AV, green closed curve).

Test IV: A fourth example is presented in Fig. 15, where four AVs are requested to serve 17 WS. The 2D environment is cluttered with static obstacles with narrow passages and three depots. Two

https://doi.org/10.1017/S0263574714002872 Published online by Cambridge University Press

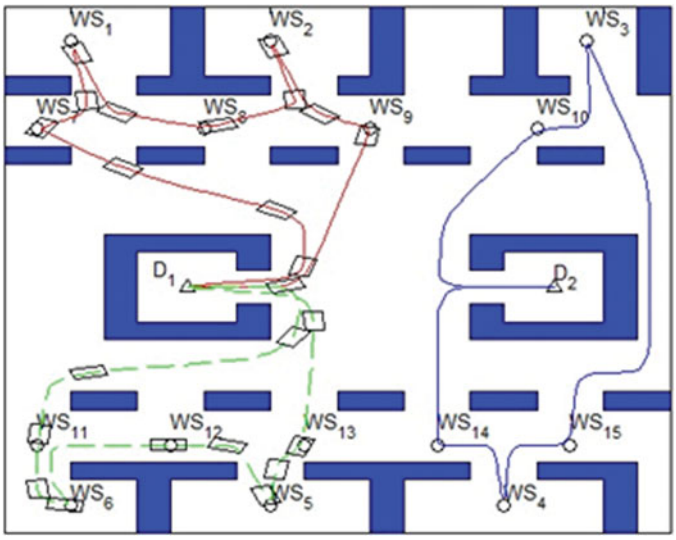


Fig. 9. The solution tours and the traces of the 1-AV and the 3-AV on their tours.

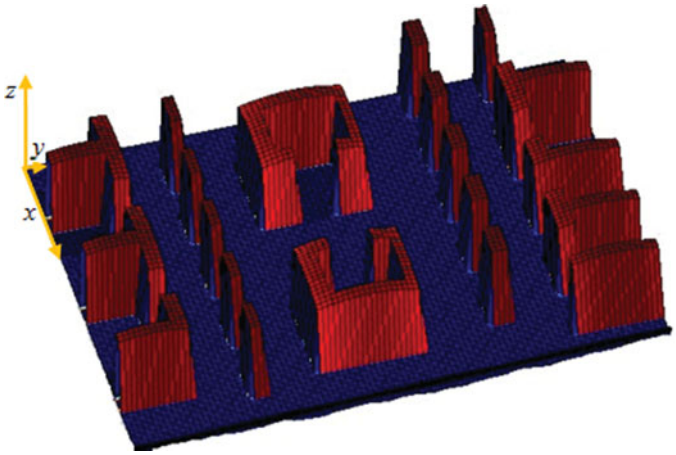


Fig. 10. The bump-surface corresponding to the industrial environment of Fig. 8.

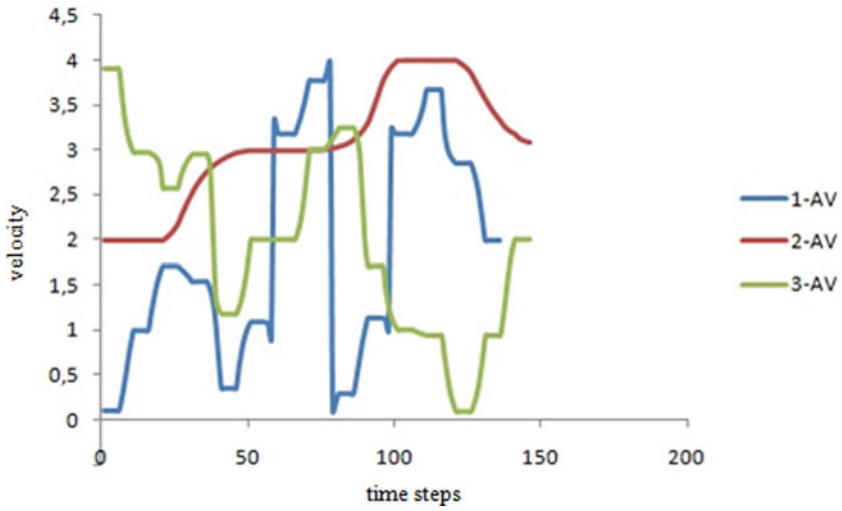


Fig. 11. The velocity profiles.

https://doi.org/10.1017/S0263574714002872 Published online by Cambridge University Press

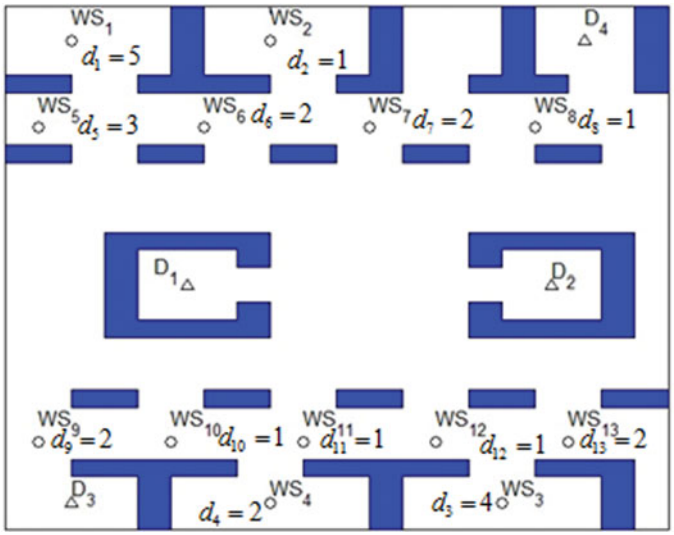


Fig. 12. A 2D environment cluttered with 13 workstations and 4 depots.

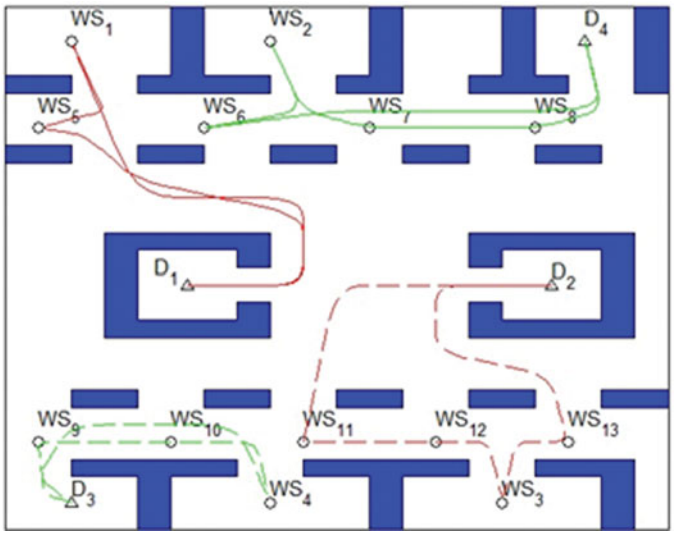


Fig. 13. The solution tours.

AVs share the same depot. The number r of control points between each pair of WS is set equal to three. All the AVs have the same capacity: $CA_i = 9$, $i = 1, ..., 4$. The proposed solution is presented in Fig. 15 and the corresponding Bump-Surface is shown in Fig. 16. It must be noted that, when the AVs share the same depot, they start their motion with a time delay determined by the proposed approach in order to avoid collisions between them. Furthermore, it is assumed that the AVs at each workstation have the ability to make a turn without violating the curvature constraint. As one can see from Fig. 15, the generated tour solutions for each vehicle are as follows:

- $D_1 \rightarrow WS_1 \rightarrow WS_2 \rightarrow WS_3 \rightarrow D_1$ (1-AV, magenta closed curve).
- $D_1 \rightarrow WS_8 \rightarrow WS_9 \rightarrow WS_{10} \rightarrow WS_{11} \rightarrow D_1$ (2-AV, black closed curve).
- $D_3 \rightarrow WS_{15} \rightarrow WS_{14} \rightarrow WS_{13} \rightarrow WS_{12} \rightarrow WS_{17} \rightarrow WS_{16} \rightarrow D_3$ (3-AV, cyan closed curve).
- $D_2 \rightarrow WS_6 \rightarrow WS_5 \rightarrow WS_7 \rightarrow WS_4 \rightarrow D_2$ (4-AV, blue closed curve).

Between the test cases I and II, an increase in the computational running time equal to 9.2%, is observed. This is caused by the increase of the number of the AVs and of the number of WS that made the complexity of scheduling problem much harder. Between the test cases II and III, an increase equal to 3.4% in computational running time is observed. This is caused by the increase of

https://doi.org/10.1017/S0263574714002872 Published online by Cambridge University Press

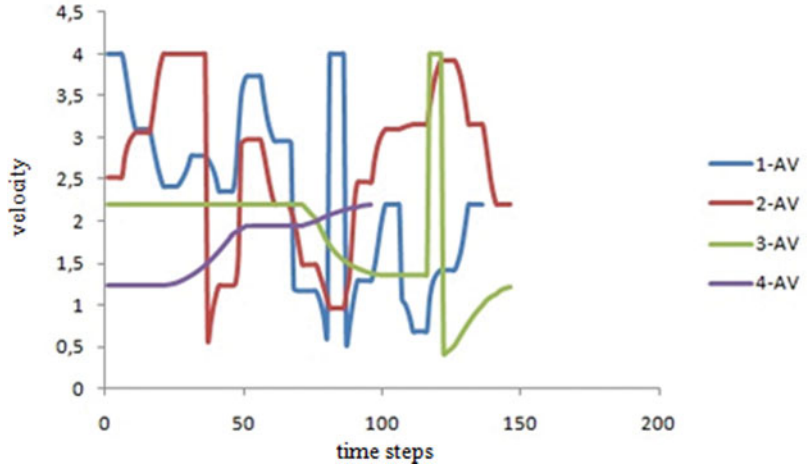


Fig. 14. The velocity profiles.

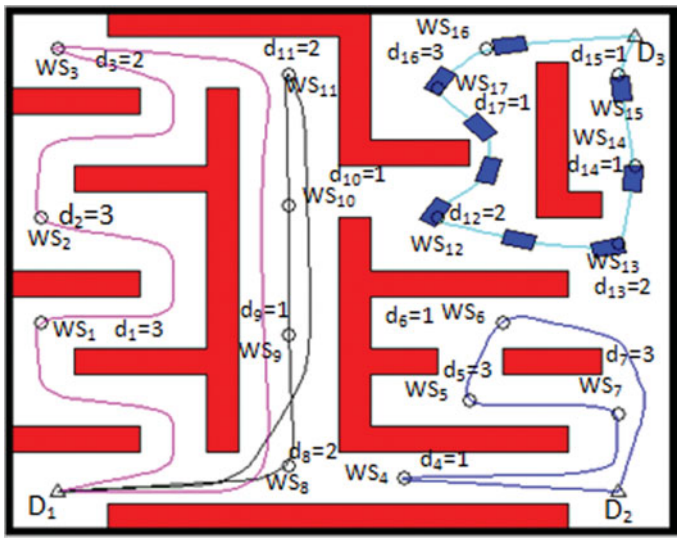


Fig. 15. The 2D environment cluttered with 3 depots and 17 WS. The colored closed curves show the motion of the AVs. Indicatively the colored rectangle shows some selected time-steps of an AV which is moving on the computed path.

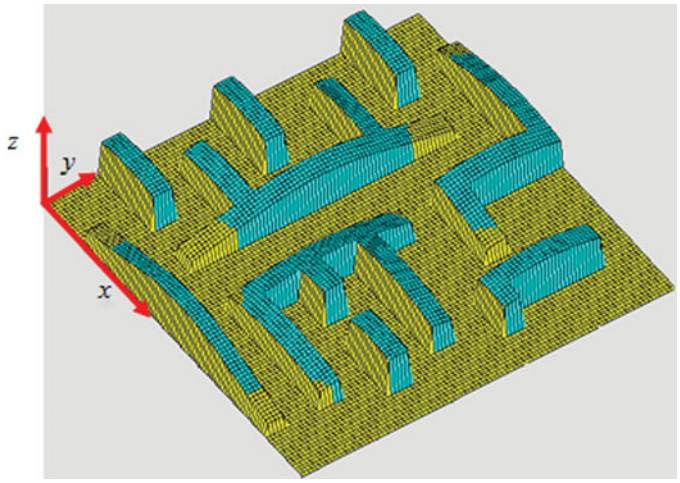


Fig. 16. The corresponding bump-surface.

https://doi.org/10.1017/S0263574714002872 Published online by Cambridge University Press

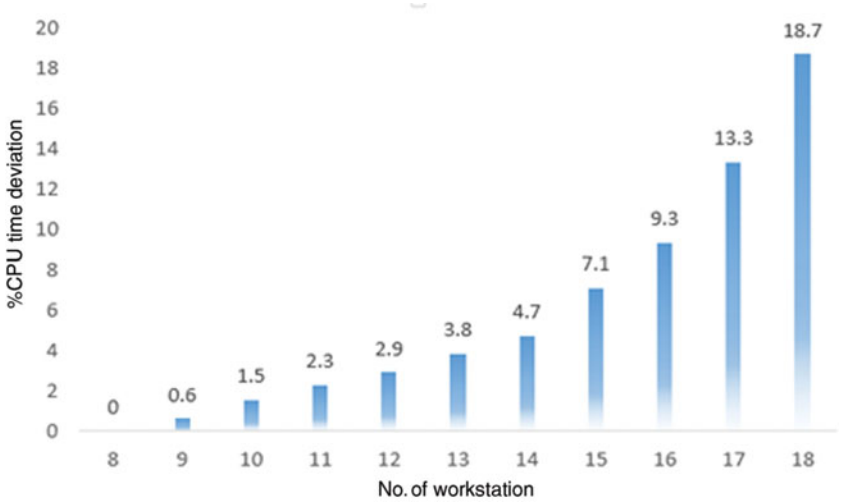


Fig. 17. The effect of WS in CPU time.

the AVs. Between the test cases III and IV, an increase equal to 15.1% in computational running time is observed. This is caused mainly by the increase of the number of the WS.

Figure 17 shows the increase of the CPU time versus the increase of the number of WS for the test case II. For comparison purposes, the percentage increase from the CPU time corresponding to 8 WS is presented, which is taken as a reference computational time. It can be concluded from Fig. 17 that the computational time increase is almost linear, but after the case of 14 WS the rate of increase is quite higher.

As one can see from the above examples, the developed GA is quite efficient at coordinating and scheduling safe motions for a fleet of AVs in complicated environments with corridors, walls, and other obstacles. Note that, the generated solution tours satisfy all the established motion planning and scheduling criteria presented in Section 2.

6. Conclusions

This paper presents a novel method for managing (coordinating) a fleet of AVs used for logistics operations in indoors factory environments. The objective is to determine the best vehicles routes so that all the WS in the shop floor are served at the lowest possible cost. The proposed method is based on the Bump-Surface concept in order to express the combined problem of VPPSP as a global constrained optimization problem. In order to resolve the derived optimization problem, a GA is proposed which utilizes a complex chromosome consisting of integer- and real-value parts. Experiments are conducted showing the effectiveness of the proposed method.

Future work will be concentrated on applying the proposed concept in more complicated environments, where a set of AVs are requested to serve a set of WS providing pickup and delivery tasks while moving safely (i.e., avoiding any collision with obstacles) in a partially known environment. In addition, transferring the method from the simulation level to the heart of an actual logistics system is a significant issue for a possible future work.

References

1. P. N. Azariadis and N. A. Aspragathos, "Obstacle representation by Bump-surfaces for optimal motion-planning," *Robot. Auton. Syst.* **51**(2–3), 129–150 2005.
2. S. Berman, E. Schechtman and Y. Edan, "Evaluation of automatic guided vehicle systems," *Robot. Comput.-Integr. Manuf.* **25**(3), 123–126 2008.
3. J.-H. Chen and S.-Y. Ho, "A novel approach to production planning of flexible manufacturing systems using an efficient multi-objective genetic algorithm," *Int. J. Mach. Tools Manuf.* **45**(7–8), 949–957 2005.
4. D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning* (1st Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1989, ISBN:0201157675).

https://doi.org/10.1017/S0263574714002872 Published online by Cambridge University Press

5. D. Herrero-Pérez and H. Martínez-Barberá, "Modeling distributed transportation systems composed of flexible automated guided vehicles in flexible manufacturing systems," *IEEE Trans. Ind. Inform.* **6**(2), 166–180 2010.
6. Y. K. Hwang and N. Ahuja, "Gross motion planning – a survey," *ACM Comput. Surv.* **24**(3), 219–291 1992.
7. M. S. LaValle, *Planning Algorithms* (Cambridge University Press New York, NY, USA, 2006, ISBN:0521862051).
8. A. C. Nearchou, "The effect of various operators on the genetic search for large scheduling problems," *Int. J. Prod. Econ.* **88**(2), 191–203 2004.
9. T. Nishi and R. Maeno, "Petri net decomposition approach to optimization of route planning problems for AGV systems," *IEEE Trans. Autom. Sci. Eng.* **7**(3), 523–537 2010.
10. L. E. Parker, "Path Planning and Motion Coordination in Multiple Mobile Robot Teams," **In**: *Encyclopedia of Complexity and System Science* (R. Meyers, ed.) (Springer, 2009).
11. L. Piegl and W. Tiller, *The NURBS Book* (Springer-Verlag, Berlin, Heidelberg, 1997).
12. K. Ravi Raju and O. V. Krishnaiah Chetty, "Addressing design and control issues of AGV-based FMSs with Petri net aided simulation," *Comput. Integr. Manuf. Syst.* **6**(2), 125–134 1993.
13. M. Solomon, "Algorithms for the vehicle routing and scheduling problem with time windows constraints," *Oper. Res.* **35**, 254–265 1987.
14. J. N. Tsitsiklis, "Special cases of travelling salesman and repairman problems with time windows," *Networks* **22**, 263–282 1992.
15. J. Xiao and Z. Michalewicz, "An Evolutionary Computation Approach to Robot Planning and Navigation," **In**: *Soft Computing in Mechatronics* (K. Hirota and T. Fukuda, eds.) (Springer-Verlag, Heidelberg, Germany, 2000) pp. 117–128.
16. E. K. Xidias and P. N. Azariadis, "Mission design for a group of autonomous guided vehicles," *Robot. Auton. Syst.* **59**(1), 34–43 2011.
17. E. K. Xidias, A. C. Nearchou and N. A. Aspragathos, "Vehicle scheduling in 2D shop floor environments," *Industrial Robot* **36**(2), 176–183.
18. X. Jin and A. Ray, "Navigation of autonomous vehicles for oil spill cleaning in dynamic and uncertain environments," *Int. J. Control* **87**(4), 787–801 2014.

Appendix

$A\widehat{V}$	The set of autonomous vehicles
j-AV	The j-autonomous vehicle
M	The overall number of vehicles
G	The overall number of depots
$\widehat{D}$	The set of depots
D_j	The j-depot
$W\widehat{S}$	The set of work stations
WS_i	The i-work station
N	The overall number of work stations
CA	The capacity quantity
$v_j(s)$	The velocity of the j-AV
$v^{\max}$	The maximum velocity
d_i	The delivery quantity associated with *i-work stations*
$\boldsymbol{W}$	The normalized workspace
$\mathbf{p}_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j}) \in [0, 1]^3$	The net of points defined in the 3D Euclidean space
N_g	The grid size
S	The bump-surface
$N_i^2(u)$ and $N_j^2(u)$	The b-spline base functions
$\mathbf{R}_j(s)$	The j-tour associated with the j-AV
$\mathbf{p}_i^j$	The control points which define the $\mathbf{R}_j(s)$
K^j	The overall number of control points $\mathbf{p}_i^j$
ω_j^i	Weight factors
$b^j = r(N_j + 1)$	The overall number of the intermediate control points
N_j	The total number of $W\widehat{S}^j$
r	The number of intermediate control points between each pair of WS

https://doi.org/10.1017/S0263574714002872 Published online by Cambridge University Press

$\boldsymbol{g}_l^j$	The intermediate control points
L_j	The arc length of the image of the $\mathbf{R}_j(s)$onto
$\mathbf{a}_j^k$	A set of vertices in the perimeter of the j-AV
N	The overall number of $\mathbf{a}_j^k$
H_j^k	The "flatness" of the image $\mathbf{S}_z(\mathbf{a}_j^k(s))$ of $\mathbf{a}_j^k(s)$ on S
Q_j^i	The region occupied by the j-AV
d_j	The diameter of the circle centered at the center of j-AV
d	The distance between the centers of two vehicles j-AV and j'-AV
E_j	A penalized length function
τ_0	A constant which depends on the friction between the wheels and the ground
g	The gravity constant
Δt_j^i	The infinitesimal time
t	The time required for j-AV to travel along $\mathbf{R}_j(s)$
E_j^{int}	The internal energy of $\mathbf{R}_j(s)$
$C_j(s)$	The curvature along the $\mathbf{R}_j(s)$
$C_j^{\max}$	The maximum curvature

https://doi.org/10.1017/S0263574714002872 Published online by Cambridge University Press