PARTICLE SWARM OPTIMIZATION-BASED COORDINATION OF A GROUP OF CONSTRUCTION VEHICLES

N. M. Kwok*, Q. P. Ha, V. T. Ngo and S. M. Hong
Faculty of Engineering
University of Technology, Sydney
Broadway, NSW 2007, Australia
*email: ngai.kwok@eng.uts.edu.au

Abstract: The problem of coordinating construction vehicles in formations is tackled using the particle swarm optimization algorithm. Coordinated multi-vehicles are anticipated to outperform single vehicles in terms of economy and flexibility. A possible approach is to apply control theoretic methods to vehicle coordination but this may require complicated system models to be employed. The particle swarm optimization (PSO) algorithm adopted in this work, as an evolutionary computation based methodology, is able to provide a near-optimal solution without the need for a precise system model. The control commands for the vehicles are treated as particles in a swarm and a sequence of controls is derived to achieve the desired formation. With regard to inter-vehicle collision avoidances, behaviour-based control strategies are incorporated into the formation framework. Simulations for multi-vehicle formation in a construction site scenario are conducted to illustrate the effectiveness of the proposed approach.

Keywords: Multi-vehicle coordination, particle swarm optimization, behaviour-based control, construction site automation.

1. INTRODUCTION

As human civilisation develops, more and more civil structures are being built at an ever increasing pace. The application of robotics in automating the construction process may become an important component in the future development of construction technologies. Tasks assigned to and performed by robots during a construction process can be broadly classified into floor finishing, board installation, exterior painting, material handling and delivery, to name a few [1]. Mobile robots or vehicles are very attractive in these application paradigms and, in particular, multi-vehicles will outperform a single robot when operating in the coordinated manner in terms of flexibility and economy.

One of the critical issues in deploying vehicles in construction sites rests on their navigational abilities [2], particularly, with severe spatial constraints and this naturally leads to the need for formation controls [3]. With this regard, control theoretic [4][5], behaviour-based [6] and A*-based searching [7] approaches have been applied. In the former approach, control designs would be very challenging as precise descriptions or models of the system are required, e.g., in a close-form or differentiable expression, and special consideration may be needed to account for numerical instabilities. On the other hand, the latter approaches require expert design knowledge and behaviours are mostly determined in a problem dependent manner.

Alternative methodologies for path planning or formation control, such as soft-computing and evolutionary computation free from the above mentioned burdens, have increasingly drawn the attentions of researchers and engineers in recent years. For example, a neural network, applying the self-learning principle, was employed in the floor coverage problem scenario [8] in constructions.

However, the coordination of multi-vehicles was not within the scope of the research work described there. Fuzzy logic based approaches for vehicle path planning were reported in [9] for cases of navigation of a single vehicle in construction sites but multi-vehicle coordination issues were also not addressed. In [10], the Ant system algorithm was employed for its natural representation of vehicles as ants and the major focus of the work therein was on the allocation of job schedules with regard to vehicle coordinations. Furthermore, the multi-agent concept implemented with genetic algorithms (GA) was reported in [11][12], where the problem domain addressed was on roadside following for vehicle navigation. In essence, vehicles were treated as living species evolving by adaptation to natural selections imposed by the constraints from the kerb boundaries. However, one of the hurdles in applying the GA is the determination of the many control parameters, e.g., selection schemes and crossover/mutation probabilities, during the algorithmic design stage.

From an alternative point of view, the collective motion of vehicles can be treated as an aggregation of microscopic particles evolving through the solid and liquid phases [13] with a balance between diffusion and cohesion. This analogy thus inspires another form of evolutionary computation, namely, the particle swarm optimisation (PSO) algorithm [14] for its simplicity and promising performance in wide application areas. The principle of PSO is based on the exchange of social knowledge and personal experiences among the individuals (particles) in the swarm. Specifically, the algorithm operates by coding potential solutions as individual particles and simulates bird flocks or fish schools moving across the solution space. A solution is produced as the best particle in the swarm. The PSO algorithm has been used in robot path planning [15], navigation [16] and many other design optimisation problems.

In this work, the PSO algorithm is employed in the multi-vehicle formation control paradigm. This method will be applied for the coordination of the motions of construction vehicles, assuming the available information of the locations of the vehicles, in a construction site where there are severe spatial constraints. Speed and steering commands for the vehicles are derived from the PSO algorithm in order to establish the required formation. Moreover, to avoid inter-vehicle collisions during the formation initialization, behaviour-based control schemes are applied for its ease in design and promising performances.

The rest of the paper is structured as follows. In Section 2, the vehicle coordination problem is briefly reviewed and the particle swarm optimisation algorithm is introduced in Section 3. In Section 4, the proposed approach is developed and simulation results are presented in Section 5. Finally, a conclusion is drawn in Section 6.

2. VEHICLE COORDINATION CONTROL

The problem scenario considered is that construction vehicles are deployed for material delivery, navigating through constrained spaces and in task dependent formations such as lines, columns or wedges and others. Furthermore, the formation parameters may be time-varying as the task requires.

Let the motion of the vehicles under coordination control be described by the following motion model,

$$x^i_{k+1} = f(x^i_k, u^i_k)$$

$$\begin{bmatrix} x^i_{k+1} \\ y^i_{k+1} \\ \phi^i_{k+1} \end{bmatrix} = \begin{bmatrix} x^i_k + v^i_k T \cos(\phi^i_k + \gamma^i_k T) \\ y^i_k + v^i_k T \sin(\phi^i_k + \gamma^i_k T) \\ \phi^i_k + \gamma^i_k T \end{bmatrix}, \quad (1)$$

where x^i_{k+1} is the state of the ith vehicle at time $k+1$ consisting of its xy-coordinate and orientation ϕ with respect to the x-axis, u^i_k is the control containing the velocity v^i_k, steering rate γ^i_k and T is the sampling time.

The goal of vehicle coordination is to derive a sequence of controls for each vehicle, i.e.,

$$u = \{u^i_0, \cdots, u^i_k\} \quad (2)$$

such that the trajectories

$$x = \{x^i_0, \cdots, x^i_k\} \quad (3)$$

followed by the vehicles are attracted to the desired ones of a formation determined by a high-level path planner.

These controls can be obtained by applying control theoretic approaches [4][5], behaviour-based schemes [6] or A*-based architecture [7]. However, the need to avoid inter-vehicle collisions and obstacles may increase the system complexity, computational load and give rise to numerical instabilities. Therefore, an evolutionary computation technique, the particle swarm optimization algorithm is adopted for its simplicity and flexibility.

3. PARTICLE SWARM OPTIMIZATION

The particle swarm optimization (PSO) algorithm can be viewed as an agent based heuristic search method where potential solutions are coded as particles. The algorithm contains a recursive iteration loop (generations) and can be described by the following pseudo code.

```
1.  Initialize particles randomly across the
    solution space
2.  Set generation count to zero
3.  While not terminate
    3.1. Evaluate the particle fitness
    3.2. Find the best particle
    3.3. Find the best instances of particles
         against generations
    3.4. Calculate particle velocities
    3.5. Update particle locations
    3.6. Increase generation count
4.  Terminate if generation count expires.
```

In the context of vehicle coordination, the vehicle speeds and steering commands are coded as particles,

$$u^i_{p,k} = \{v^i_{p,k}, \gamma^i_{p,k}\}^T, i = 1, \cdots I, p = 1, \cdots P, \quad (4)$$

where p is the particle index and i is the index for a vehicle.

The particles in the solution space are allowed to move with arbitrary velocities (which is distinguished from the vehicle speeds). The initial particle velocities may be all set to zero or random numbers. In subsequent generations, the velocities are determined as

$$v^i_{p,j+1} = wv^i_{p,j} + c^i_1(u^i_{g,j} - u^i_{p,j}) + c^i_2(\hat{u}^i_{p,j} - u^i_{p,j}), \quad (5)$$

where w is a weighting factor representing the momentum of the particle, $(c_1, c_2) \in [c_{min}, c_{max}]$ are uniform random numbers denoting as the gain factors, $u^i_{g,j}$ (group-best) is the particle that gives the best performance within the swarm at the jth generation, $\hat{u}^i_{p,j}$ (personal-best) is the best performing instance of the pth particle over the past generations. The momentum weighting factor determines the randomness in the search for solutions, while the gain factors are responsible for the rate of convergence to the optimum solution.

Assuming a unity sampling time step, the particle locations in the solution space are updated in the next generation as

$$u^i_{p,j+1} = u^i_{p,j} + v^i_{p,j+1}. \quad (6)$$

The algorithm then iterates through a pre-determined number of generations and the best particle $u^i_{g,\bar{j}}$ at the terminating generation $\bar{j}$ is used in controlling the ith vehicle to move to its next location from time k to $k+1$ by letting $u^i_k \leftarrow u^i_{g,\bar{j}}$.

4. PROPOSED APPROACH

In this paper, the coordination of multi-vehicles into formations is achieved by combining the PSO algorithm and the behaviour-based motion strategy in deriving the motion commands as well as avoiding inter-vehicle collisions.

4.1 Particle Structure

Let there be m vehicles to be coordinated, hence, there are m sequences of control commands to be determined by the PSO. The approach adopted assumes that a high-level path planner is available to design the required formation and each vehicle knows it current location. This gives a set of formation locations or *virtual* vehicles as

$$\mathbf{F} = \{F_1, \cdots F_m\}, \quad F_f = \{x^f, y^f\}^T, \quad f = 1, \cdots, m, \quad (7)$$

where each formation location contains its corresponding xy-coordinates and the orientations are aligned with the x-axis. On the other hand, the initial locations of the vehicles are not specified (i.e., not in a formation) but their locations are known to the PSO algorithm.

The control commands are represented by a set of *control* particles. Note that there are I vehicles each also contains a set of P *location* particles describing the possible locations of vehicles. At each time step, the control particles are used to *move* the vehicles according to the motion model (Eq. 1). The predicted vehicle locations are shown in Fig. 1.
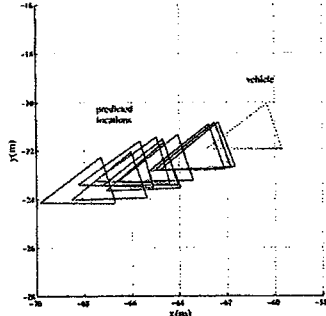

Fig. 1 – Predicted vehicle locations according to the speed and steering commands determined by particles.

4.2 Vehicle-Formation Indexing

In order to follow a formation, vehicles need to be allocated an index corresponding to the virtual vehicles in the formation. At the start of the formation with the available knowledge on the virtual formation and vehicle locations, a cross-correspondence distance table is formed by calculating the distances

$$d^{fi} = \sqrt{(x^f - x^i)^2 + (y^f - y^i)^2}, \quad (8)$$

where the superscript f denotes the formation and i denotes the vehicle. Then, for each vehicle, find the minimum distance to the virtual vehicle and assign the correspondence index.

$$i \leftrightarrow f, \text{if } \hat{d}^{fi} = \min_f \{d^{fi}\}. \quad (9)$$

By adopting this indexing scheme, the initial distances between the real and virtual vehicles are minimized. Consequently, the time taken to reach the formation may be reduced.

4.3 Formation Strategy

The formation of vehicles into platoons is treated as a tracking problem. Due to the non-holonomic constraints of the vehicles they cannot turn abruptly. First, the distances between the vehicle particles and the virtual vehicles are calculated, giving $\Delta x_p^{fi}, \Delta y_p^{fi}$ respectively. Assume that the formation is required to move along the x-axis from left to right (it is straightforward to generalize to other formation movements as it is assumed to be performed by a high-level path planner). The strategy adopted is to assign a pseudo target behind the virtual vehicle such that the vehicle chases and tracks the formation. The pseudo target is given by

$$\Delta x_p^{vi} = \Delta x_p^{fi} - 0.85 d^{fi}$$
$$\Delta y_p^{vi} = \Delta y_p^{fi} - 0.05 d^{fi} \text{sign}(\Delta y_p^{fi}), \quad (10)$$

where superscript v denotes the target vehicles. The factors 0.85 and 0.05 are scales determined experimentally. By examining the expressions above, it is revealed that the pseudo target will approach the virtual vehicle when the real vehicle is moving into the vicinity of the formation where $d^{fi} \to 0$ and, thus, provides a smooth tracking.

4.4 Particle Fitness

The PSO algorithm relies on the determination of relative fitness values among the particles where the group-best and personal-best particles are obtained. The fitness function is an aggregation of the vehicle-formation distance and the vehicle-formation angular separation. The distances were as in Eq. 8, giving d_p^{fi} on a particle basis, while the angular separation is given by

$$\theta_p^{fi} = \phi_p^i - \tan^{-1}(\Delta y_p^{vi} /(\Delta x_p^{vi} + L)), \quad (11)$$

where L is the vehicle length. Here, each angle is referred to the angular separation between a vehicle particle to that of a virtual vehicle. The fitness value for each vehicle particle is given as

$$f_p^{fi} = d_p^{fi} + \left| \theta_p^{fi} \right|. \quad (12)$$

The group-best fitness is obtained from

$$\hat{f}_g^{fi} = \max_p \{f_p^{fi}\}. \quad (13)$$

Similarly, the personal-best fitness is

$$\hat{f}_{p,j}^{fi} = \max_j \{f_{p,j}^{fi}\}, \quad (14)$$

determined from the history of the fitness of the pth particle up to the jth generation.

Finally, these fitness values are used in calculating the particle velocities and updating their locations in the solution space (see Section 3).

4.5 Control Bounds

Since the PSO is a heuristic search method, the speed and steering commands derived may be infeasible for typical vehicles that exceed their kinematic or dynamic limitations. Therefore, the control signals should be bounded or clamped. Taking this into account, the values of the control particles are assigned as

$$v^i_{p,k} \leftarrow v_{max}, \text{if } v^i_{p,k} > v_{max}$$
$$v^i_{p,k} \leftarrow 0, \text{if } v^i_{p,k} < 0. \tag{15}$$

That is, the vehicle is not allowed to travel backward in normal formation for smooth motions. Similarly, the steering command is bounded in magnitude as

$$\gamma^i_{p,k} \leftarrow \gamma_{max}, \text{if } \gamma^i_{p,k} > \gamma_{max}$$
$$\gamma^i_{p,k} \leftarrow -\gamma_{max}, \text{if } \gamma^i_{p,k} < -\gamma_{max}. \tag{16}$$

4.6 Inter-vehicle Collision Avoidance

The path required to reach the next vehicle location may introduce collisions between vehicles, since collision was not considered in the PSO routines. Therefore, a collision avoidance strategy is developed to mitigate this drawback.

For each vehicle located at x^i_k at time k, calculate the distances between other vehicles, i.e.,

$$d^{ji} = \sqrt{(x^j - x^i)^2 + (y^j - y^i)^2}, \tag{17}$$

which is a 2-dimensional array of distances. Similarly, calculate the angular separation between vehicles as

$$\theta^{ji} = \tan^{-1}\left((y^j - y^i)/(x^j - x^i)\right). \tag{18}$$

For each vehicle i, find the distance to their associated virtual formation vehicle,

$$f^{ii} = \sqrt{(x^i_f - x^i)^2 + (y^i_f - y^i)^2}. \tag{19}$$

Check for the collision condition defined as

$$(d^{ji} < 2.5L + 0.05f^{ii}) \wedge (|\theta^{ji}| < \pi/3), \tag{20}$$

where L is the length of the vehicle, $\wedge$ is the logical *and* operator. The scale factors, 2.5, 0.05 and $\pi/3$, determining the relative importance of the inter-vehicle separations in distance and orientation, are obtained experimentally. Fig. 2 shows the situation when collision occurs.
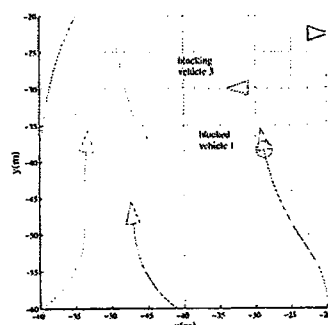

Fig. 2 – Inter-vehicle collision. Vehicle 1 blocks vehicle 3 (marked by a circle).

A potential collision is then declared between vehicles i and j when they are close to each other and one of the vehicles is in front of and blocks the other. This condition has taken into account for a dynamic threshold depending on the degree of formation completion where the risk of collision diminishes when the vehicles are in the formation, i.e., $f^{ii} \rightarrow 0$.

For the colliding vehicles, their reactive movements are frozen temporarily for a time step. Their locations become

$$x^i_{k+1} = x^i_k, \quad x^j_{k+1} = x^j_k. \tag{30}$$

4.7 Dead-lock Release

The inter-vehicle collision avoidance strategy adopted may give rise to dead-lock conditions, especially when the vehicles are moving towards each other (both in front and block the other). A release of the dead-lock condition is proposed as follows.

During the collision avoidance stage, a list of colliding vehicle index pairing is maintained. For example, let vehicles 1 and 3 are temporarily frozen, the list will read as

$$L_{fz} = \begin{cases} 1 & 3 \\ 3 & 1 \end{cases}, \tag{31}$$

which signifies that vehicle 1 blocks vehicle 3 and the reverse also holds, thus, producing a dead-lock.

If the number of paired entries in the list is more than one, the list is searched for duplicated vehicle indices. Following the above example and applying the proposed strategy, vehicle 1 will be driven backward using a random speed and a random steering angle. Furthermore, the corresponding entry in the dead-lock list is removed. The procedure then repeats for other dead-locked vehicle pairings such that multiple dead-locks are subsequently removed.

5. RESULTS

The proposed vehicle coordination framework for formation initialization and control, using PSO and the incorporation of behavioural control for collision avoidance, is applied in simulations for different types formations and each case has a different set of initial vehicle locations. The simulation conditions are listed below.

Table I - Types of formations

Case	Formation
1	Column
2	Wedge

Table II - Initial vehicle locations

Case	Initial location
1	Left hand regions in a column
2	Upper-left region randomly located
3	Lower-left region randomly located
4	Randomly located across the site boundary

Consider seven vehicles that are homogeneous with each vehicle having measures $3m \times 2m$ in length and width. Let the construction site be bounded, e.g., $\pm 70m$ in the xy-coordinates. The formation starts at the centre, $x = 0$. The initial vehicle locations are chosen arbitrarily, in accordance to the cases listed in Table II, and their orientations are also initialized randomly. The formation moves from left to right (increasing x-coordinate) and the formation is allowed to change dynamically. The last virtual vehicle (located at the most negative y-coordinate) moves in a higher speed than the first virtual vehicle.

In the column formation, trajectories of the vehicles are shown in Fig. 3a through 3d. Irrespective of the vehicle starting locations and the dynamically adjustment of the formation, the vehicles follow the formation closely. It is also illustrated that the trajectories are smooth and the formation virtual vehicles are well tracked. This observation has verified the satisfactory performances provided by the proposed PSO algorithm. The trajectories although indicate some crossovers, however, they are separated in the time domain where the inter-vehicle collision avoidance procedure has been operating effectively. Notably, as shown in the lower-left region of Fig. 3c, one of the vehicle trajectories contains a rather sharp change. This is caused by the blocked vehicle that moved backward in order to avoid inter-vehicle collision and dead-lock.

For the wedge formation case, similar levels of performances are also observed from the plots in Fig. 4. It is noted that the successful establishment of the desired formation, from the initial vehicle locations, is independent on the type of the desired formation (column vs. wedge). Furthermore, the tracking of virtual vehicles is formation-independent and the trajectory complexity is also not related to the type of formation. In general, the proposed approach can be straight forwardly extended to cases of a larger number of vehicles and inhomogeneous vehicle sizes.

6. CONCLUSION

A particle swarm optimisation (PSO) algorithm has been proposed for the formation control of vehicles deployed in construction sites. Sequences of vehicle speed and steering commands are derived from the PSO that can drive the vehicles to follow near-optimal trajectories leading to a formation. An interesting feature here is that the formation parameters can be time-varying. The collisions between vehicles are avoided by using a behaviour-based strategy. Simulation results for various initial vehicle locations and formations have demonstrated the effectiveness of the proposed approach.

ACKNOWLEDGEMENTS

REFERENCES

[1] A. Warszawski and R. Navon, "Implementation of robotics in building: current status and future prospects," *Journal of Construction Engineering and Management*, January/February 1998, pp. 31-41.

[2] S. Lee, T. M. Adams and B. Ryoo, "A fuzzy navigation system for mobile construction robots," *Automation in Construction*, vol. 6, 1997, pp. 97-107.

[3] J. P. Desai, J. P. Ostrowski and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *IEEE Trans. on Robotics & Automation*, vol. 17, no. 6, pp. 905-908, Dec. 2001.

[4] J. Jongusuk and T. Mita, "Tracking control of multiple mobile robots: a case study of inter-robot collision-free problems," *Proc. IEEE Intl. Conf. on Robotics & Automation*, Seoul, Korea, May 2001, pp. 2885-2890.

[5] R. Fierro, A. K. Das, V. Kumar and J. P. Ostrowski, "Hybrid control of formations of robots," *Proc. IEEE Intl. Conf. on Robotics and Automation*, Seoul, Korea, May 2001, pp. 157-162.

[6] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Trans. on Robotics & Automation*, vol. 14, no. 6, Dec. 1998, pp. 926-939.

[7] V. T. Ngo, A. D. Nguyen, Q. P. Ha, "Toward a generic architecture for robotic formations planning and control," *Proc. The Sixth International Conference on Intelligent*, Phuket, Thailand, Dec. 2005, pp. 89-96.

[8] S. X. Yang and C. Luo, "A neural network approach to complete coverage path planning," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 34, no. 1, Feb, 2004, pp. 718-725.

[9] A. R. Soltani and T. Fernando, "A fuzzy based multi-objective path planning of construction sites," *Automation in Construction*, vol. 13, 2004, pp. 717-734.

[10] Y. Ding, Y. Han and J. Jiang, "Multi-robot cooperation method based on the ant algorithm," *Proc. 2003 IEEE Swarm Intelligence Symposium*, Apr. 2003, pp. 14-18.

[11] H. Chen and Z. Xu, "Path planning based on a new genetic algorithm," *Proc. Intl. Conf. on Neural Networks and Brain*, Oct. 2005, pp. 788-792.

[12] Z. Cao, M. Tan, S. Wang, Y. Fan and B. Zhang, "The optimization research of formation control for multiple mobile robots," *Proc. 4th World Congress on Intelligent Control & Automation*, Shanghai, China, June 2002, pp. 1270-1274.

[13] G. Gregorie, H. Chate and Y. Tu, "Moving and staying together without a leader," *Physica D*, vol. 181, 2003, pp. 157-170.

[14] M. Clerc and J. Kennedy, "The particle swarm – explosion, stability and convergence in a multidimensional complex space," *IEEE Trans. on Evolutionary Computation*, vol. 6, no. 1, Feb. 2002, pp. 58-73.

[15] Y. Qin, D. Sun, N. Li and Y. Cen, "Path planning for mobile robot using the particle swarm optimization with mutation operator," *Proc. 3rd Intl. Conf. on Machine Learning & Cybernetics*, Shanghai, China, Aug. 2004, pp. 2473-2478.

[16] S. Doctor and G. K. Venayagamoorthy, "Unmanned vehicle navigation using swarm intelligence," *Proc. IEEE Intl. Conf. on Intelligent Sensing & Information Processing*, Chennai, India, Jan. 2004, pp. 249-253.
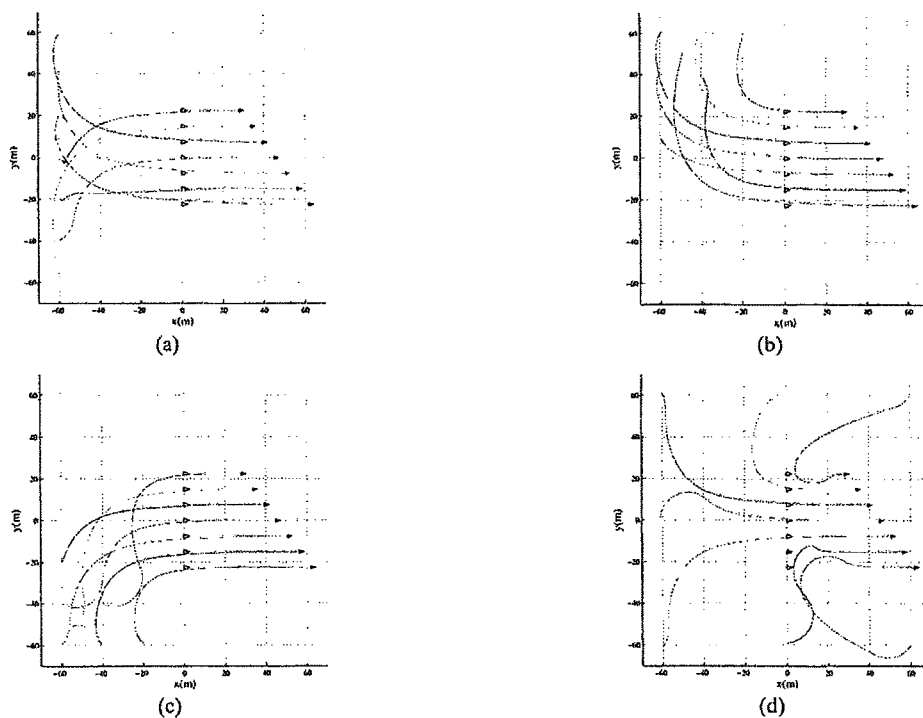
Fig. 3 – Results for COLUMN formation. (a) Vehicles started from left, (b) vehicles started from upper-left region, (c) vehicles started from lower-left region, (d) vehicles started in distributed regions. The formation changes from their initial form (shown as empty triangles) with increasing vehicle speeds to the final formation (shown in black triangles).
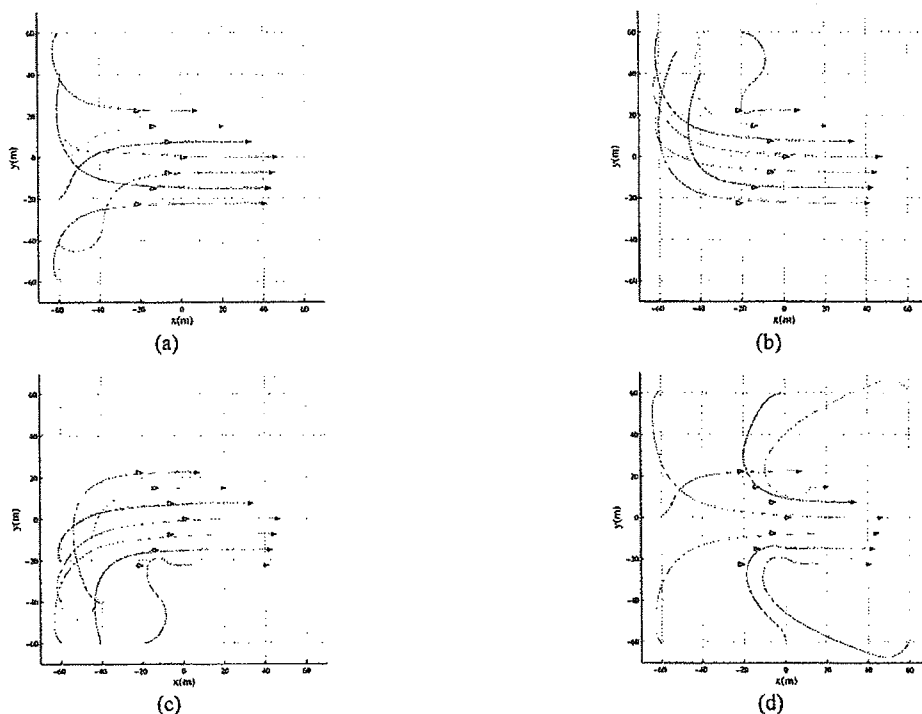

Fig. 4 – Results for WEDGE formation. (a) Vehicles started from left, (b) vehicles started from upper-left region, (c) vehicles started from lower-left region, (d) vehicles started in distributed regions. The formation changes from their initial form (shown as empty triangles) with increasing vehicle speeds to the final formation (shown in black triangles).