

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Path Planning for Formations of Mobile Robots using PSO Technique

Martin Macaš¹, Martin Saska², Lenka Lhotská¹,
Libor Přeučil¹ and Klaus Schilling²

¹Dep. Of Cybernetics, Czech Technical University, ² University of Wuerzburg
¹Czech Republic, ²Germany

1. Introduction

Multi-robotics systems are currently subject of major interest in the robotics literature. In the leading journals can be found hundreds of articles, published in the last few years, concerning applications and theoretical studies of small groups maintaining in fixed formation (Fua et al., 2007; Kaminka et al., 2008) as well as swarms of thousands robots (Derenick & Spletzet, 2007; Daigle et. al, 2008).

The large systems can be moreover represented by work published in (Peasgood et al., 2008) where collision free trajectories to reach individual goals are designed for 100 robots. The method using graph and spanning tree representation is developed for utilization in underground mine environment. In another example (Kloetzer & Belta, 2007), a large swarm of robots is controlled using hierarchical abstractions. Inter-robot collision avoidance and environment containment are there guaranteed applying centralized communication architecture. Finally work presented in (Milutinovi & Lima, 2006) applies a Stochastic Hybrid Automation model for modeling and control of multi-agent population composed of a large number of agents. In this method probabilistic description of task allocation as well as distribution of the population over the work space is considered. As an example of common multi-robots application highway traffic coordination can be mentioned. In (Pallotino et al., 2007) is presented decentralized approach using traffic rules for control of tens vehicles. The method enables dynamically adding and removing of the vehicles and is based only on local communication which makes the algorithm scalable.

Algorithms designed for smaller groups of robots are usually aimed at maintaining of vehicles in a predefined formation for the purpose of cooperative tasks accomplishing (as can be e.g. box pushing (Vig & Adams, 2006), load carrying (Tanner et al., 2003), snow shoveling (Saska et al., 2008) or aircraft as well as satellites cooperative mapping (Ren & Beard, 2003; Kang & Sparks, 2000). Another interesting application of formation driving is presented in (Fahimi, 2007) where autonomous boats are maintained in formations under sliding mode, which provides faster movement. The hot research topics in formations of autonomous robots, investigated nowadays, include e.g. data fusion: (Kaminka et al., 2008) represents the sensing capabilities using a monitoring multigraph. This approach allows the robots to adjust to sensory failures by switching of control graphs on-line. An application of data fusion can be cooperative localization of mobile formations: (Mourikis & Roumeliotis,

2006b) addresses a problem of resource allocation which provides the sensing frequencies, for each sensor on every robot, required in order to maximize the positioning accuracy of the group. This work is extended by a performance analysis providing upper bound on the robots' expected positioning uncertainty which is determined as a function of the sensors' noise covariance and relative position measurements (Moutikis & Roumeliotis, 2006a). Another separate branch of the research relevant to the formations of mobile robots is solving how to achieve the desired formation. An approach considering this task without assigning specific configuration to specific robots is published in (Kloder & Hutchinson, 2006) where a new representation for the configuration space of permutation-invariant multi-robot groups is described.

This chapter is focussed on the path planning and formation driving of autonomous car-like robots. In the literature formation driving approaches are divided into the three main groups: virtual structure, behavioral techniques, and leader-following methods. In the virtual structure approaches the entire formation is regarded as a single structure where to each vehicle is given a set of control to follow the desired trajectory of formation as a rigid body (Beard et al., 2001; Lalish et al., 2006). In behavior based methods the desired behaviors are designated for each agent and the final control is derived as a weighted sum with respect to the importance of each task (for basic ideas see (Langer et al., 1994; Parker, 1998)). These classical methods have been extended for maintaining of shape of formations using desired patterns (Lawton et al., 2003; Balch & Arkin, 1998). In the leader-following approaches, a robot or even several robots are designated as leaders, while the others are following them (Desai et al., 2001; Das et al., 2003). Example of the methods using multiple leaders is presented in (Fredslund & Mataric, 2002) where due to limited communication the followers are led by their closest neighbors. Unfortunately all these results are focused on the following of a leader's trajectory which is assumed as an input of the methods. It is supposed that the trajectory is designed by a human operator or by a standard path planning method modified for formation requirements. In the literature there is no adequate method providing flexible control inputs for the followers as well as designing an optimal path for the leader of formation responding to the environment which is necessary for fully autonomous systems.

This chapter proposes a path planning approach developed for leader-following formations of car-like robots which is an extension of work (published by the authors' team in (Saska et al., 2006)) designed for single robot. In this extended method a reference path calculated by the leader should be feasible for all following robots without changing a relative distance in the formation. This requirement can be satisfied using a solution which is composed of smoothly connected cubic splines and can be calculated on-line. Qualities of the result like the length and minimal radius of the resulting path as well as the distance to obstacles are merged into a discontinuous penalty function.

The resulting global minimization problem is solved with Particle Swarm Optimization (PSO). Since the original PSO scheme has been developed, many various modifications were proposed that more or less improve the method. In context of our optimization problem, we are strongly limited by the requirement on low time complexity. Therefore, every modification that could be used here must not lead to any slow down of the convergence. This fact suspends some sequential hybridization of PSO and any other optimization technique. Also, any sub-swarm based and multi-start algorithms are not suitable. It will be shown that the original global-best PSO performs well and even significantly better than

genetic algorithms. Nevertheless, the chapter shows some comparison of PSO with limited maximum velocity and constricted PSO that can improve the result in case of small swarm and number of iterations.

2. Formation Control

The formation driving method described in this section is based on a leader-follower approach, in which the followers should follow a leader's trajectory. The method was developed by Barfoot and Clark (Barfoot et al., 2002; Barfoot & Clark, 2004) and later improved for following of trajectories with arbitrary shape within our team (Saska et al., 2006; Hess et al., 2007). In this chapter there will be published only the parts of formation control necessary for understanding of restrictions applied in the path planning while a detailed description of control inputs for each vehicle can be found in (Saska et al., 2006; Hess et al., 2007).

In the description of the method as well as in the final experiments, known map of environment and utilization of car-like robots with limits for maximum velocity v_r and minimum turning radius R_r will be assumed. Furthermore, around each vehicle will be considered distance d_r from its center in which the obstacles have to be avoided (d_r is usually a function of robot's width).

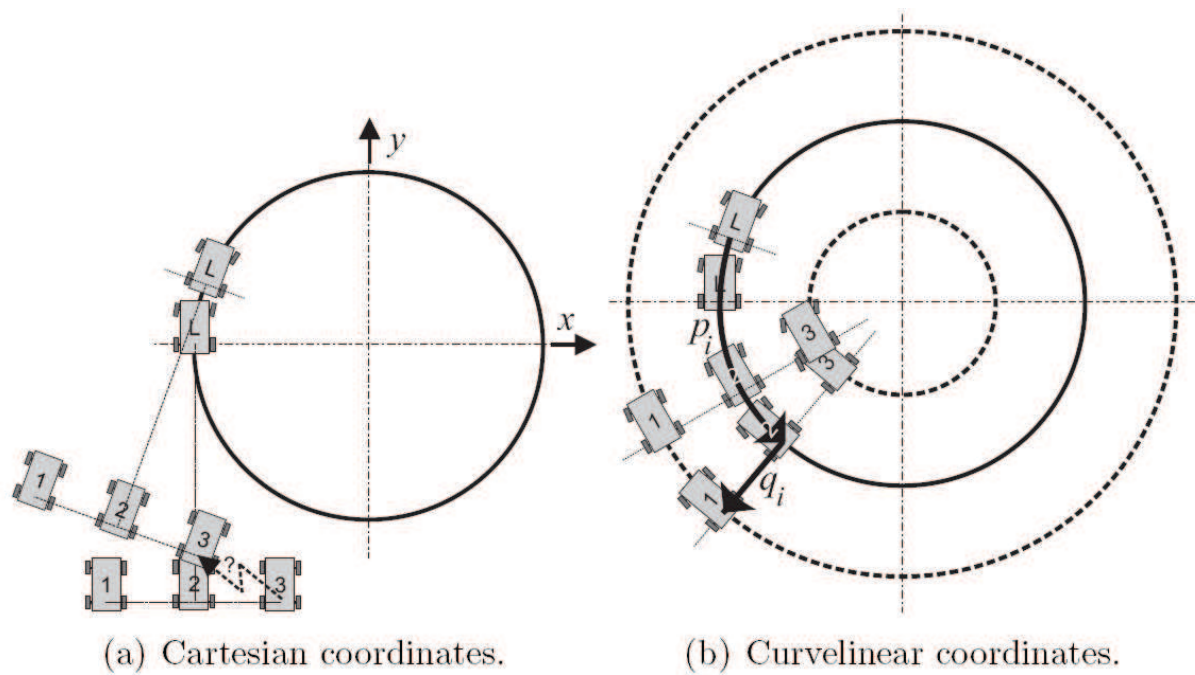


Figure 1. Two subsequent snapshots of formation driving using fixed position of followers in Cartesian (a) and Curvilinear (b) coordinates. Solid lines denote path of leader while paths of followers are denoted by dashed lines

Important fact of the formation driving of car-like robots that needs to be considered is caused by impossibility to change heading of the robot on spot. Due to this feature formations with fixed relative distance in Cartesian coordinates cannot be used, because

such structure makes smooth movement of the followers impossible (simple example is shown in Fig. 1). Therefore we utilized an approach in which the followers are maintained in relative distance to the leader in curvilinear coordinates with two axes p and q , where p traces movement of leader and q is perpendicular to p as is demonstrated in Fig. 1b. The positive direction of p is defined from actual position of the leader back to the origin of its movement and the positive direction of q is defined in the left half plane in direction of forward movement.

The shape of formation is then uniquely determined by states $\psi_L(t_{p_i(t)})$ in travelled distance $p_i(t)$ from actual position of the leader along its trajectory and by offset distance $q_i(t_{p_i(t)})$ between positions of the leader and the i th follower in perpendicular direction from the leaders' trajectory. The parameters $p_i(t)$ and $q_i(t)$ defined for each follower i can be varying during the mission and $t_{p_i(t)}$ is time when the leader was at the travelled distance $p_i(t)$ behind the actual position. $\psi_L(t) = \{x_L(t), y_L(t), \Theta_L(t)\}$ denotes the configuration of a leader robot at time t , and similarly $\psi_i = \{x_i(t), y_i(t), \Theta_i(t)\}$, with $i \in \{1, \dots, n_r\}$, denote the configuration for each of the n_r follower robots at time t . The Cartesian coordinates x_t, y_t for an arbitrary configuration $\psi(t)$ define the position of a robot and $\Theta(t)$ denotes its heading.

To convert the state of the followers in curvilinear coordinates to the state in rectangular coordinates $\psi_i(t)$ the following equations can be applied:

$$\begin{aligned} x_i(t) &= x_L(t_{p_i(t)}) - q_i(t_{p_i(t)}) \sin(\theta_L(t_{p_i(t)})) \\ y_i(t) &= y_L(t_{p_i(t)}) + q_i(t_{p_i(t)}) \cos(\theta_L(t_{p_i(t)})) \\ \theta_i(t) &= \theta_L(t_{p_i(t)}), \end{aligned} \quad (1)$$

where $\psi_L(t_{p_i(t)}) = \{x_L(t_{p_i(t)}), y_L(t_{p_i(t)}), \Theta_L(t_{p_i(t)})\}$ is state of the leader in time $t_{p_i(t)}$. Applying the leader following approach using p, q coordinates we can easily determine inadmissible interval of turning radius for the leader of formation as $R_f^-(t); R_f^+(t)$, where

$$\begin{aligned} R_f^-(t) &= \max_{i=1, \dots, n_r} (-R_r + q_i(t)) \\ R_f^+(t) &= \min_{i=1, \dots, n_r} (R_r + q_i(t)). \end{aligned} \quad (2)$$

These restrictions must be applied due to the different turning radius of the robots on the different position in the formation during turning. It is obvious that the robot following inner track should go slower and with smaller turning radius than the robot further from the centre of turning.

Since the leader trajectory has to be collision free for the leader but also for the followers, the shape of the formation should be included to the avoidance behaviour. The extended obstacle free distance for the leaders' planning can be then expressed as

$$d_f(t) = d_r + \max_{i=1,\dots,n_r} |q_i(t)|. \quad (3)$$

Remark 2.1 Time dependence and asymmetry of the formation will be for simplification of the algorithm description omitted and the variables will be considered as constants:

$$d_f = \max_{0 < t < T} d_f(t),$$

$$|R_f| = \min\left(\min_{0 < t < T} R_f^+(t), \min_{0 < t < T} -R_f^-(t)\right) \quad (4)$$

where T is total time of the formation movement.

3. Path Description and Evaluation

The path planning for the leader of formation can be realized by a search in the space of functions. In this approach the space is reduced to a sub-space which only contains strings of cubic splines. The mathematic notation of a cubic spline (Ye & Qu, 1999) is

$$g(s) = As^3 - Bs^2 + Cs + D, \quad (5)$$

where s is within the interval $<0;1>$ and A, B, C, D are constants. The whole string of the splines is then in 2D case uniquely determined by $8n$ variables (n denotes the amount of splines in the string). The initial and desired state (position and orientation) of the formation is specified by 8 equations, while continuity of first and second derivative in the whole path, which is important for the formation driving as is shown in (Saska et al., 2006), is guaranteed by $6(n-1)$ equations.

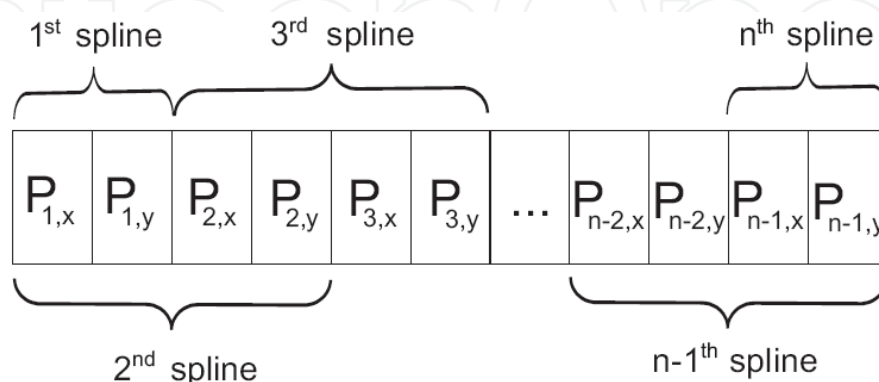


Figure 2. Path representation

Therefore, only $2(n-1)$ degree of freedom define the whole path, which conforms to positions of the points in the spline connections. The whole path representation used in our method is shown in Fig. 2.

Each solution achieved by the global optimization method is evaluated by a cost function. The global minimum of this function corresponds to a smooth and short path that is safe (there is sufficient distance to obstacles). The cost function was in introduced method used in the form

$$f = f_{length} + f_{distance} + f_{radius}. \quad (6)$$

where part f_{length} corresponds to the length of the path which in 2D case can be computed by

$$f_{length} = \int_0^1 \sqrt{(g'_x(s))^2 + (g'_y(s))^2} dt. \quad (7)$$

The component $f_{distance}$ (Fig. 3a) penalizes the paths close to an obstacle and it is defined by equation

$$f_{distance} = \begin{cases} d^{-2}, & \text{if } d_f < d \\ d^{-2} + p_{df}, & \text{if } d_r < d < d_f \\ d^{-2} + p_{df} + p_{dr}, & \text{if } d < d_r \end{cases} \quad (8)$$

where p_{df} penalizes solutions with a collision that can be avoided by a change in the formation and p_{dr} penalizes paths with a collision of the leader. Parameter d denotes minimal distance of the path to the closest obstacle and can be expressed as

$$d = \min_{o \in O} \min_{t \in \langle 0;1 \rangle} \|o - g(s)\|, \quad (9)$$

where O is set of all obstacles in the workspace of the robots.

The part of the cost function f_{radius} (Fig. 3b), that is necessary because of using the car-like robots as well as due to presented formation driving approach, is computed according

$$f_{radius} = \begin{cases} r^{-2}, & \text{if } R_f < r \\ r^{-2} + p_{rf}, & \text{if } R_r < r < R_f \\ r^{-2} + p_{rf} + p_{rr}, & \text{if } r < R_r \end{cases} \quad (10)$$

where solutions penalized only by p_{rf} can be repaired by a formation changing, while paths with radius smaller than R_r do not meet even requirements for a single robot. Parameter r is minimal radius along the whole path and it is defined by

$$r = \min_{t \in \langle 0;1 \rangle} \frac{\left| \begin{matrix} g'_x(s) & g'_y(s) \\ g''_x(s) & g''_y(s) \end{matrix} \right|}{(g'_x(s)^2 + g'_y(s)^2)^{\frac{3}{2}}}. \quad (11)$$

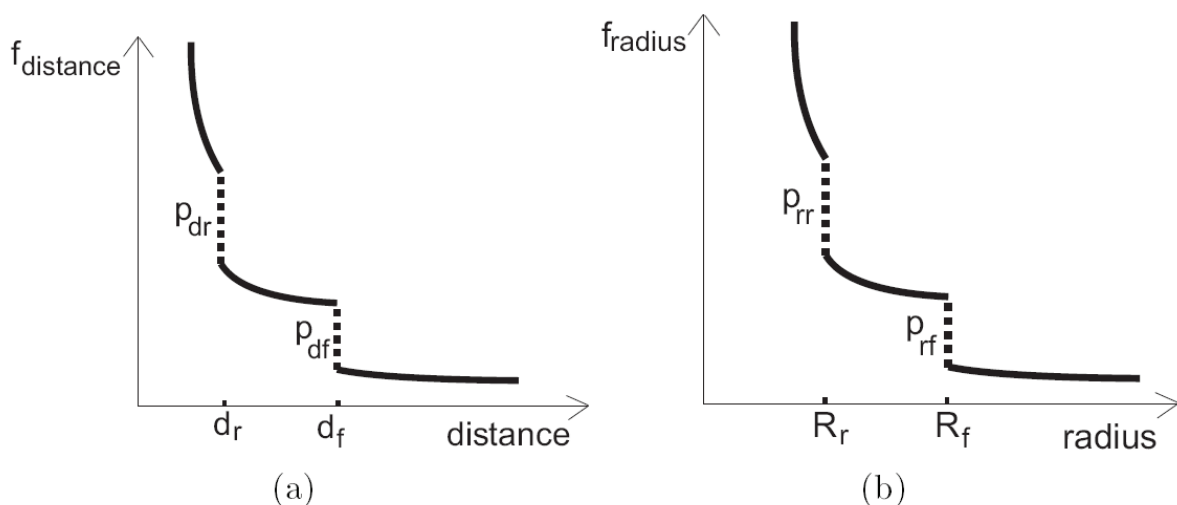


Figure 3. (a) $f_{distance}$, (b) f_{radius} - components of cost function with denoted penalizations

4. Particle Swarm Optimization

Each particle i is represented as a D -dimensional position vector $\vec{x}_i(t)$ and has a corresponding instantaneous velocity vector $\vec{v}_i(t)$. The position vector encodes robot path according to the schema depicted in Fig. 2. In our simple case of three splines (two spline connections), the position vector is 4-dimensional and $\vec{x}_i(t) = \{P_{1,x}, P_{1,y}, P_{2,x}, P_{2,y}\}$. Furthermore, each particle remembers its individual best value of fitness function and position $\vec{p}_i(t)$ that has resulted in that value. During each iteration t , the velocity update rule (12) is applied on each particle in the swarm:

$$\begin{aligned} \vec{v}_i(t) = & w \vec{v}_i(t-1) + \\ & + \varphi_1 R_1 (\vec{p}_i - \vec{x}_i(t-1)) + \\ & + \varphi_2 R_2 (\vec{p}_g - \vec{x}_i(t-1)). \end{aligned} \quad (12)$$

The $\vec{p}_g(t)$ is the best position of the entire swarm and represents the social knowledge. Another alternative can be "local best PSO", where the best position from a local neighborhood is used instead of $\vec{p}_g(t)$. We chose the "global-best PSO" because of faster convergence that is consistent with our requirement on low time complexity. The parameter w is called inertia weight and during all iterations decreases linearly from $w_{start}=0.8$ to $w_{end}=0$. The symbols R_1 and R_2 represent the diagonal matrices with random diagonal elements drawn from a uniform distribution between 0 and 1. The parameters φ_1 and φ_2 are scalar constants that weight influence of particles' own experience and the social knowledge. The parameters were set $\varphi_1 = \varphi_2 = 2$ in compliance with literature recommendation.

Next, the position update rule (13) is applied:

$$\vec{x}_i(t) = \vec{x}_i(t-1) + \vec{v}_i(t) \quad (13)$$

If any component of $\vec{v}_i(t)$ is less than $-V_{max}$ or greater than $+V_{max}$, the corresponding value is replaced by $-V_{max}$ or $+V_{max}$, respectively. The V_{max} is maximum velocity parameter. This parameter (as well as the velocity and position vectors) is related to the spatial dimensions of the planning area. For the area with 80000×40000 pixels, some preliminary tests showed that $V_{max} = 3000$ was suitable setting.

The update formulas (12) and (13) are applied during each iteration and the $\vec{p}_i(t)$ and $\vec{p}_g(t)$ values are updated simultaneously. The algorithm stops if maximum number of iterations is achieved.

There are some specific moments in our application. The swarm initialization is the most important one. The particular components of the particle positions have the direct interpretations. They are coordinates of 2-D points in the robot workspace. Therefore, it is suitable to initialize the position vectors into a rectangle with one corner in the start position and the opposite corner in the goal position. However, there can be some different initialization strategies (e.g. initializing the spline connection points over the whole workspace or on the line connecting the start and the goal position).

For our particular scenarios, we choose the initial position to be uniform random numbers from $\langle 30000; 40000 \rangle$. The same initialization was used for genetic algorithm described below.

5. Genetic Algorithm

The PSO has been compared to the most commonly used nature-inspired method - genetic algorithm (GA). In all experiments, the same GA scheme with stochastic uniform selection, scattered crossover and gaussian mutation was used (Vose, 1999). The particular settings have been chosen experimentally.

In the stochastic uniform selection a line is laid out in which each parent corresponds to a section of the line of length proportional to its scaled value. The algorithm moves along the line in equal sized steps and allocates a parent from the section it lands on. The "scattered" crossover selects randomly the genes to be recombined. The Gaussian mutation adds a random number drawn from Gaussian distribution with zero mean and variance linearly decreasing from $0.5r_0$ to $0.125r_0$, where r_0 is the the initial range (for our experiments, $r_0 = 40000 - 30000 = 10000$). Moreover, elitism has been used that copies two best individuals from the previous generation into the new generation if a better individual was not created in the new generation. This prevents the loss of best solution and accelerates the convergence.

6. Implementation Details

Great number of evaluations is required by available optimization methods and therefore computational complexity of the cost function is key factor for real time applications. The most calculation-intensive part of the equation (6) is $f_{distance}$. It is done by big amount and complexity of obstacles from which the distance needs to be computed. In the presented method a distance grid map of the environment is pre-computed. Each cell in such matrix denotes minimum distance of relevant place to the closest obstacle according to equation (8). The regions outside the polygon denoting walls of the building or inside the obstacles could be signed by infinite value, because they are infeasible for the formation movement. Nevertheless due to the simple initialization used in this chapter all particles in the initial swarm can be intersecting an obstacle and therefore evaluated by the same value $f_{distance} = \infty$.

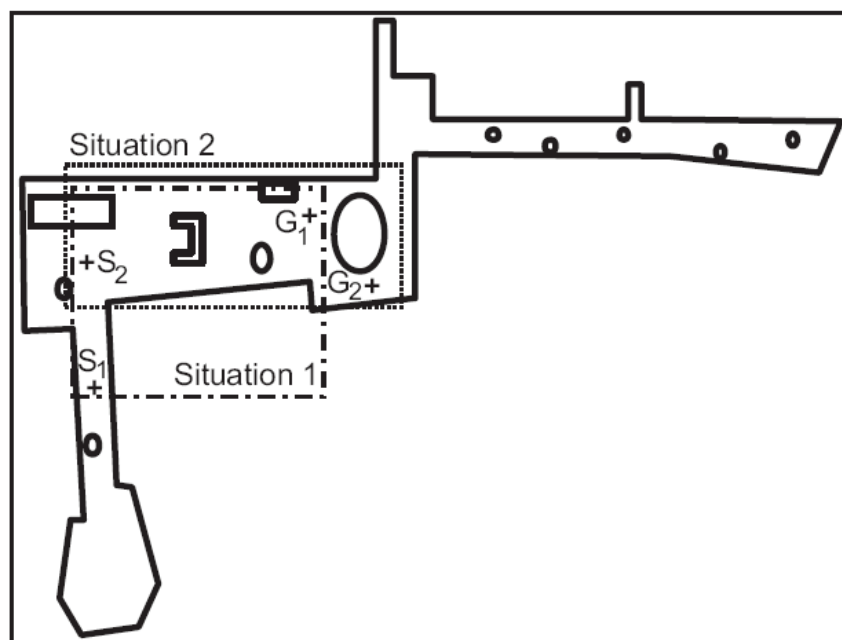


Figure 4. Map of utilized workspace with denoted zoomed areas of the scenarios: Situation 1 and Situation 2

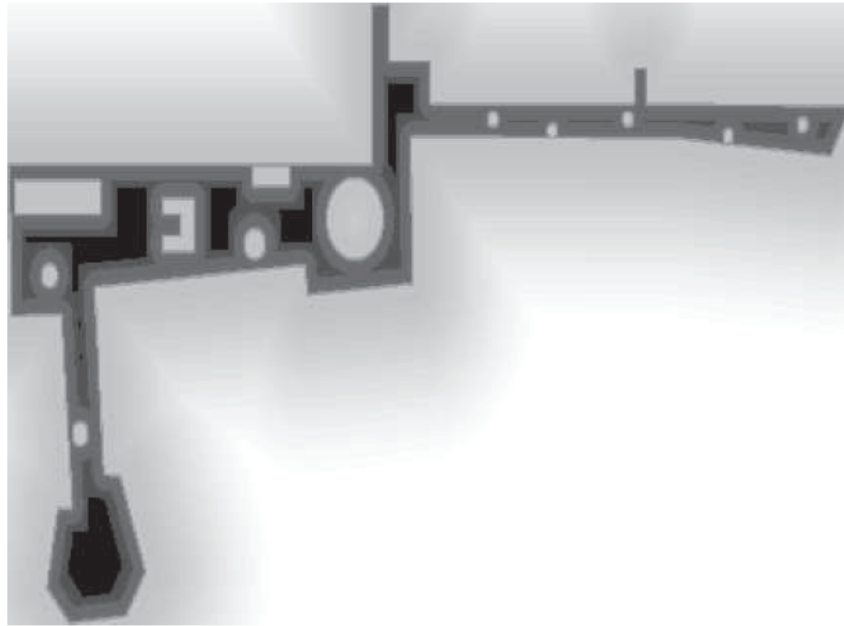


Figure 5. Distance map used for computing of the cost function. Black color denotes the regions where $f_{distance} = 0$ and white color denotes the region with maximum values of $f_{distance}$

In such case even the smartest optimization method degrades to a random search. A solution could be to artificially add a rising of $f_{distance}$ outside the polygon from the walls of building (similarly inside the circular obstacles the increase will be from the borders to the center of obstacle) which enables the optimization method to reach the feasible space. Big advantage of such grid-map approach is possibility to use obstacles with arbitrarily complicated shape, that is usually done by autonomous mapping technique. An occupancy grid that is obtained by a range finder can be used as well. An example of the robot workspace with obstacles that was used for experiments is depicted on the Fig. 4 and the appropriate distance map is drawn in the Fig. 5.

7. Experiments

This section summarizes various types of experiments in static environment for two scenarios (Situation 1 and Situation 2) depicted in Fig. 4. First, the results obtained by PSO are discussed and further, the PSO is compared to genetic algorithm. The presented tests have been realized in the environment of computer science building in Wuerzburg (map is depicted in Fig. 4) which is frequently used for hardware experiments of indoor mobile robots.

7.1 PSO Results

Parameters of the PSO method were adjusted in agreement with (Saska et al., 2006), where the algorithm was used in similar application. As the test scenario were chosen situations with several local extremes corresponding to feasible as well as unfeasible paths for the leader. In Fig. 6 are presented two solutions of the Situation 1 designed by PSO method. The path evaluated by cost $f=13.02$ is close to the global optimal solution and is feasible for the

formation maintaining fixed shape. Contrariwise the second path ($f=28.71$) is close to one of the local optimal solutions and it is feasible only for a single robot. For the formation driving it means that the shape of the formation must be temporarily changed during the passage around the obstacles as well as in the loop replacing sharp unfeasible curve next to the corner of the room. We should note that the loop was created automatically by the path planning method. Such manoeuvres could be useful e.g. in crossroads of narrow corridors where straightforward movement is impossible due to the restriction of turning radius.

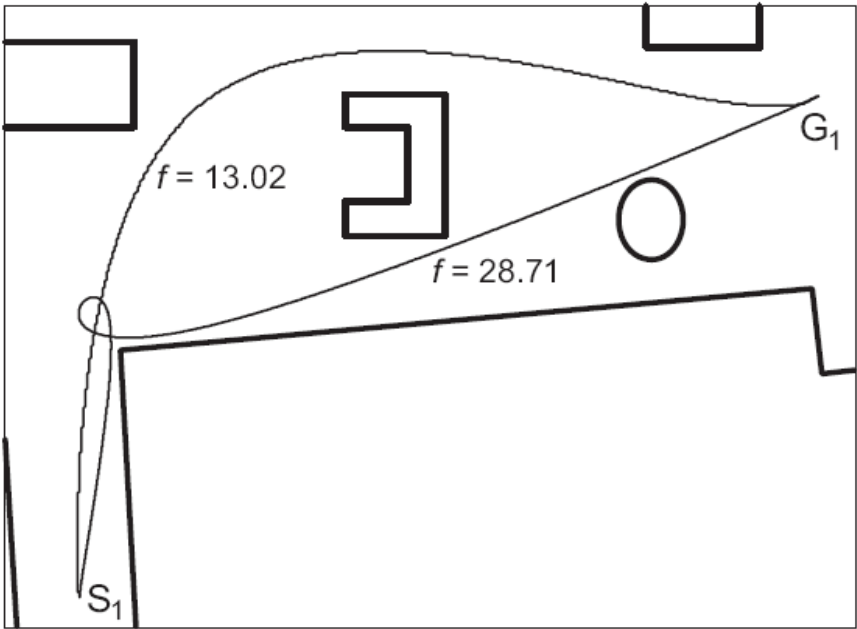


Figure 6. Two different solutions of Situation 1 obtained by PSO

Results of the second scenario, Situation 2, are shown in Fig. 7 where the solution with $f=13.82$ is feasible for the complete formation whereas the second solution ($f=18.31$) requires small changes of the positions of outer followers. The second path is shorter than the first solution, which is close to the global minimum of the cost function (6). Therefore the second solution could be preferred in the application where the shape of formation can be easily modified.

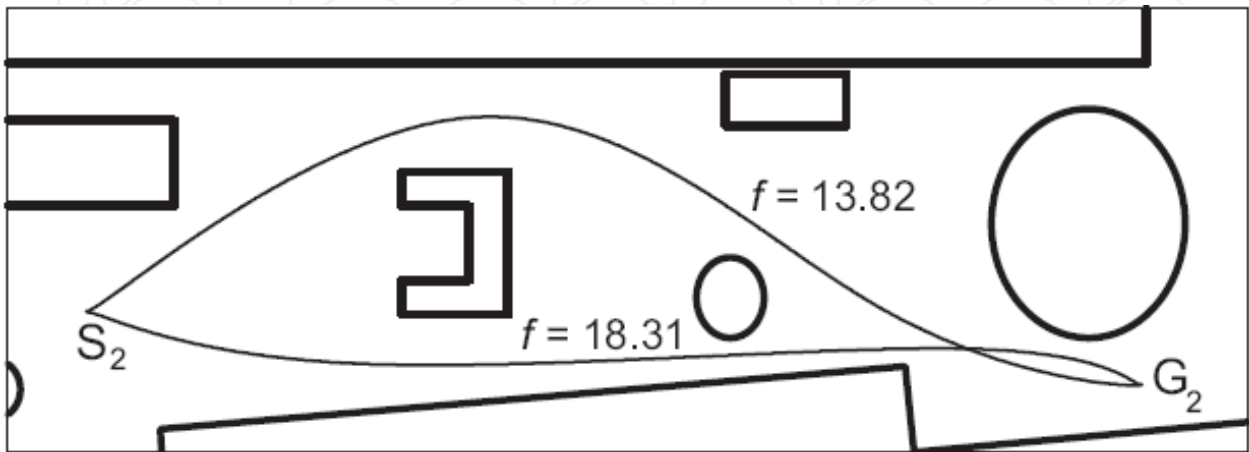


Figure 7. Two different solutions of Situation 2 obtained by PSO

7.2 Comparison with GA

The two scenarios described above were used for the comparison of PSO and GA. For both methods, the swarm (population) size was 30 and the number of iterations (generations) was 300. Such an excessive number of cost function evaluations enables better evaluation of results and the chance of the algorithm to converge into an optimum.

	min	mean	std	max
PSO	13.00	18.05	7.74	32.6
GA	13.01	39.41	155.33	1181.4

Table 1. The minimum, mean, standard deviation and maximum of the set of minimum cost values found by particular runs for Situation 1. Set of results from 100 repeated runs was used

$f_{min} \in$	$(-\infty; 14 >$	$(14; 40 >$	$(40; 300 >$	$(300; +\infty)$
PSO	70	30	0	0
GA	80	16	2	2

Table 2. Situation 1 - absolute occurrences of different values of final f_{min} in the set of 100 results of independent runs

Because of statistical purposes, 100 runs of each method (with different random initialization) has been launched. The main quality criterion used is the minimum cost function f_{min} found at a particular moment. First, we took final values of the minimum cost value found in particular runs. Basic statistical properties computed from the 100 runs are depicted in Table 1. Although the mean best PSO solutions is lower than the mean best GA solution, the difference is not statistically significant (two-sample t-test with significance level 0.05 was used to investigate the significance of difference between the methods). However, high standard deviation and high maximum (worst result) obtained for GA results shows that in some runs, the genetic algorithm found extremely poor result that do not belong to any of the two optima shown in Fig. 6. This is especially evident from the Table 2, where the histogram of best solutions is depicted. The second column corresponds to the global minimum of cost function that lies under the value $f=14$. The numbers are absolute occurrences (of totally 100 runs) of the final minimum fitness values that are lower than 14. The third column describes hits to the local optima (that lies somewhere around 28). The other two columns correspond to quite poor (probably unusable) solutions. One can observe that for the Situation 1 the GA finds these bad solutions in 4 of totally 100 cases. On the other hand the PSO always finds at least the local minimum and is more susceptible to getting stuck in the local optimum. This fact is probably a tax on the faster convergence. The higher convergence rate of PSO can be observed from Fig. 8b, where the mean temporal evolution of f_{min} is depicted. One can see that the curve for PSO decreases and reaches minimum much more rapidly than the curve measured for GA.

The results for Situation 2 are similar. This time, the mean result for GA is significantly worse (Table 3). In histogram (Table 4 and Fig. 9a), one can observe that GA again was

unable to find neither the global optimum ($f < 16$) nor the local optimum ($f \in (16;19)$) in 5 of totally 100 cases. Moreover, it found the global optima fewer times than the PSO. Again, the convergence of PSO is much faster for the Situation 2.

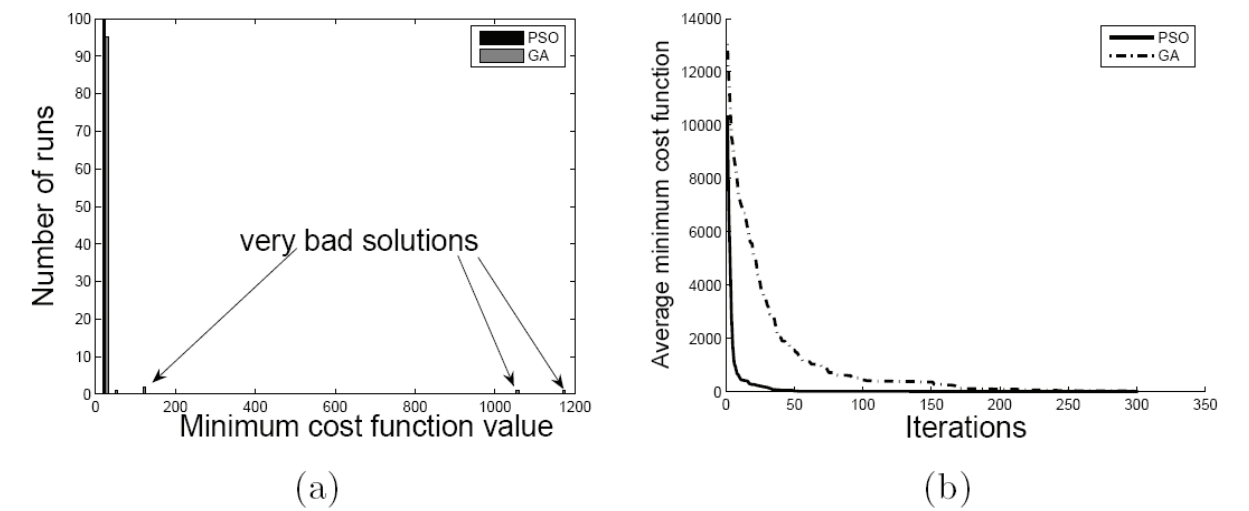


Figure 8. The results for Situation 1. The histogram of final f_{\min} values obtained from 100 runs (a) and the temporal evolution of f_{\min} values averaged over 100 runs (b)

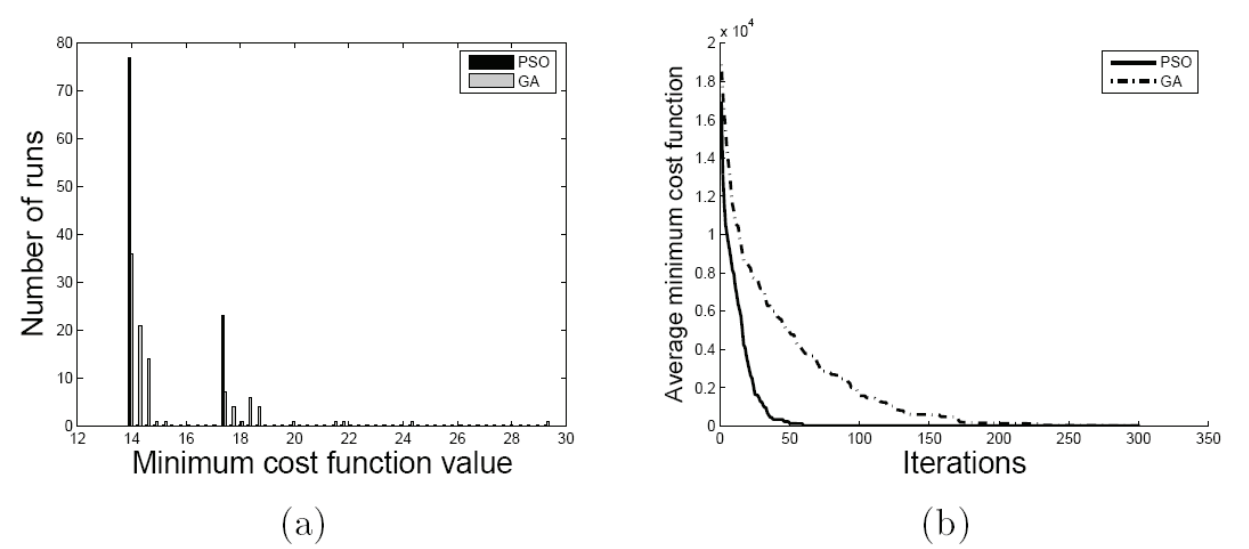


Figure 9. The results for Situation 2. The histogram of final f_{\min} values obtained from 100 runs (a) and the temporal evolution of f_{\min} values averaged over 100 runs (b)

	min	mean	std	max
PSO	13.80	14.66	1.53	17.53
GA	13.87	15.49	2.54	29.45

Table. 3 The minimum, mean, standard deviation and maximum of the set of minimum cost values found by particular runs for Situation 2. Set of results from 100 repeated runs was used

$f_{min} \in$	$(-\infty; 16 >$	$(16; 19 >$	$(19; +\infty >$
PSO	77	23	0
GA	73	22	5

Table 4. Situation 2 - absolute occurrences of different values of final *fmin* in the set of 100 results of independent runs

The conclusion of this section is that the PSO finds the solution much faster than GA . Moreover, the GA sometimes produces unusable poor solution. Both the PSO and GA parameters were tuned experimentally in some preliminary testing.

7.3 Constriction and Dynamic Inertia Weight

It has been already mentioned above that an acceptable modification of the PSO method must be very simple and should lead to improvement of algorithm's convergence rate. In this section, two very simple modifications are compared to the basic PSO described in Section 4. The first modification is the PSO with constriction coefficient (CCPSO) and the second is PSO with adaptive dynamic inertia weight (AIWPSO) (Fan & Chang, 2007). The constriction coefficient was derived from an eigenvalue analysis of swarm dynamics (Clerk, 1999). The method is used to balance exploration and exploitation trade-off. The velocity update Equation (12) is modified:

$$\begin{aligned} \vec{v}_i(t) = \chi[&w \vec{v}_i(t-1) + \varphi_1 R_1(\vec{p}_i - \vec{x}_i(t-1)) + \\ &+ \varphi_2 R_2(\vec{p}_g - \vec{x}_i(t-1))] \end{aligned}$$

(14)

where χ is the constriction coefficient, which is computed from values of φ_1 and φ_2 . We used $\varphi_1 = \varphi_2 = 2.1$ and

$$\chi = \frac{2}{|2 - \varphi - \sqrt{(\varphi^2 - 4\varphi)}|},$$

(14)

where $\varphi = \varphi_1 + \varphi_2$. The advantage is that the velocity clamping does not need to be used. The second modification - PSO with adaptive dynamic inertia weight (AIWPSO) (Fan & Chang, 2007) is based on dynamically changing inertia weight $w = w(t)$. The principal modification is the nonlinear modification of the inertia weight. The nonlinear function is given by: $w=(d)^rw_{start}$, where d is the decrease rate and has been set experimentally to $d = 0.95$ and r changes through time according to the following rules: 1. $r \leftarrow r + 1$ if the best cost function value (minimal value in the swarm) decreased (improved) and 2. $r \leftarrow r - 1$ if the best cost function value increased or remained the same. This mechanism wishes to make particles fly more quickly toward the potential optimal solution, and then through decreasing inertia weight to perform local refinement around the neighbourhood of the optimal solution.

The comparison has been done using the Situation 2 described above. The results of 100 runs are depicted on Fig. 10. For all runs, only 15 particles and 100 iterations were used. The results show that in average, better results are obtained by CCPSO, although the CCPSO has

slower convergence than PSO. On the other hand, the difference of the final best solutions is not significant. The PSO reached the global optimum area ($f < 20$) in 73 runs and the final minimum cost value averaged over 100 runs was 73.25. The CCPSO reached the global optimum area in 71 runs and the final minimum cost value averaged over 100 runs was 117.62. One can also see that the AIWPSO did not perform well. It found the global optimum area in 54 runs and the final minimum cost value averaged over 100 runs was 1159.00.

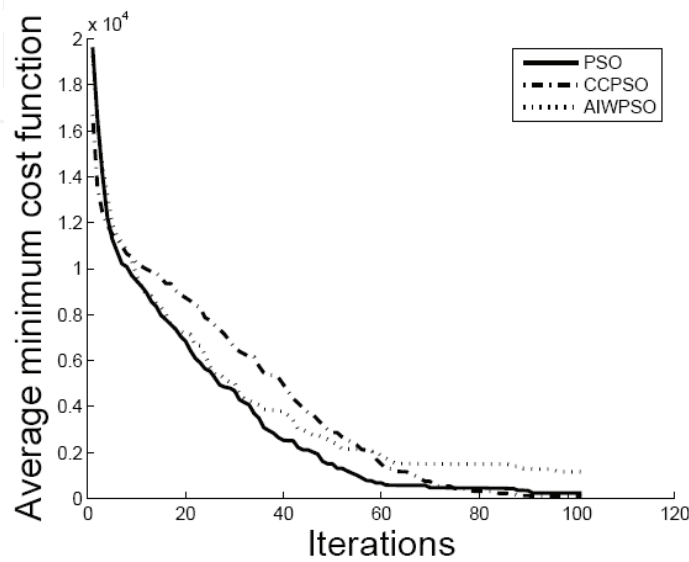


Figure 10. Comparison of three PSO modifications. The temporal evolution of f_{\min} values averaged over 100 runs

7.4 Simulation of Formation Driving

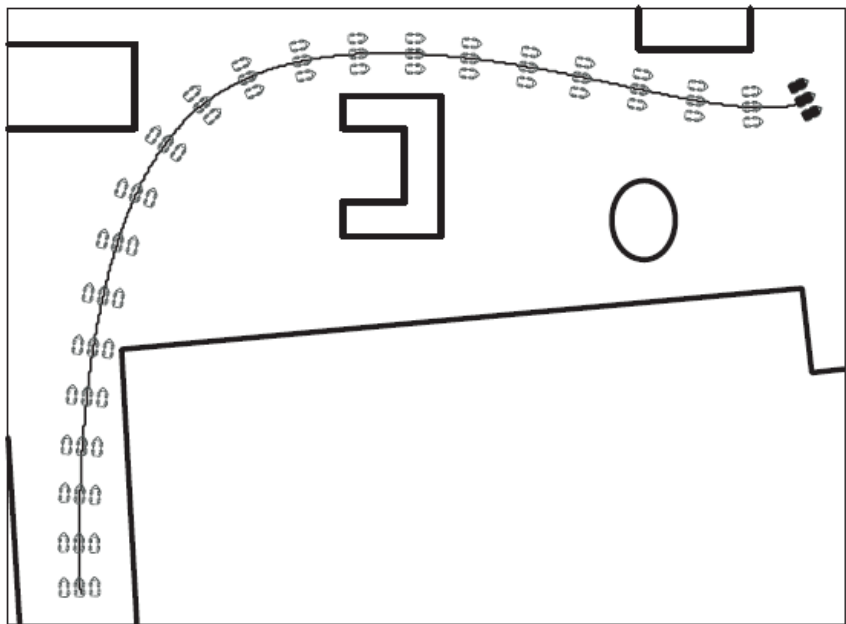


Figure 11. Simulation of formation movement

For demonstration of the formation movement was chosen the feasible solution designed by PSO in situation 1. The path planning as well as the formation driving were adjusted for the formation of three robots in the line that is perpendicular to the leader's path. In the Fig. 11 is zoomed part of the workspace with delineated positions of the robots during task execution. Robots were controlled by an approach that was presented by our team in (Hess et al. 2007).

8. Conclusion and Future Work

This chapter gave concrete recommendations about the use of PSO based spline-planner. Namely, a suitable PSO method with recommended parameter values is resumed and its main advantages and disadvantages are critically discussed. The original PSO with velocity clamping and linearly decreasing inertia weight performed well and was able to find better solution in shorter time than genetic algorithm. Because of strong limitations on time consumption, we do not recommend any complex modification. Among the two tested modification, the PSO with constriction coefficient could compete with the original PSO version. Finally, it was shown, how problematic is the use of PSO for formation path planning. In our cases, the only feasible paths corresponded to global optima of the cost function. A promising future direction is the modified random initialization of the swarm that can be adjusted in number of ways. The good initialization is simple instrument for improving the speed of the planning process that is for real time planning and dynamical environment response crucial.

9. Acknowledgments

The research was supported by the research program No. MSM6840770012 Transdisciplinary Research in the Area of Biomedical Engineering II of the CTU in Prague, sponsored by the Ministry of Education, Youth and Sports of the Czech Republic.

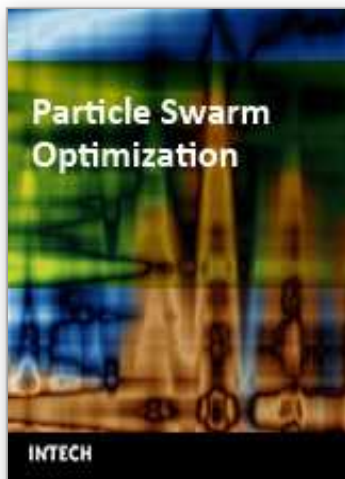
10. References

- Balch, T., & Arkin, R. C. (1998, December). Behaviour-based formation control for multi-robot teams. *IEEE Transactions on Robotics and Automation*, 14(6), 926–939.
- Barfoot, T. D., & Clark, C. M. (2004, February). Motion planning for formations of mobile robots. *Robotics and Autonomous Systems*, 46, 65–78.
- Barfoot, T. D., Clark, C. M., Rock, S. M., & D'Eleuterio, G. M. T. (2002, october). *Kinematic path-planning for formations of mobile robots with a nonholonomic constraint*.
- Beard, R., Lawton, J., & Hadaegh, F. (2001, November). A coordination architecture for spacecraft formation control. *IEEE Transactions on Control Systems Technology*, 9(6), 777 - 790.
- Clerc, M. (1999). The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization. In *Proceedings of the ieee congress on evolutionary computation* (Vol. 3, pp. 1951– 1957).
- Daigle, M. J., Koutsoukos, X. D., & Biswas, G. (2007, April). Distributed diagnosis in formations of mobile robots. *IEEE Transactions on Robotics*, 23(2), 353 - 369.

- Das, A., Fierro, R., Kumar, V., Ostrowski, J., Spletzer, J., & Taylor, C. (2003, October). A vision-based formation control framework. *IEEE Transactions on Robotics and Automation*, 18(5), 813-825.
- Derenick, J., & Spletzer, J. (2007, December). Convex optimization strategies for coordinating large-scale robot formations. *IEEE Transactions on Robotics*, 23(6), 1252-1259.
- Desai, J., Ostrowski, J., & Kumar, V. (2001, December). Modeling and control of formations of nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 17(6), 905-908.
- Fahimi, F. (2007, June). Sliding-mode formation control for underactuated surface vessels. *IEEE Transactions on Robotics*, 23(3), 617 - 622.
- Fan, S.-K. S., & Chang, J.-M. (2007). A modified particle swarm optimizer using an adaptive dynamic weight scheme. In *Hcii* (12) (p. 56-65). Springer.
- Fredslund, J., & Mataric, M. (2002, October). A general algorithm for robot formations using local sensing and minimal communication. *IEEE Transactions on Robotics and Automation*, special issue on *Advances in Multi-Robot Systems*, 18(5), 837-846.
- Fua, C., Ge, S., Duc Do, K., & Lim, K.-W. (2007). Multirobot formations based on the queue-formation scheme with limited communication. *IEEE Transactions on Robotics*, 23(6), 1160-1169.
- Hess, M., Saska, M., & Schilling, K. (2007, September). *Enhanced motion planning for dynamic formations of nonholonomic mobile robots*.
- Kaminka, G., Schechter-Glick, R., & Sadov, V. (2008, April). Using sensor morphology for multirobot formations. *IEEE Transactions on Robotics*, 24(2), 271 - 282.
- Kang, W., Xi, N., & Sparks, A. (2000). *Formation control of autonomous agents in 3d workspace*.
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings international conference on neural networks IEEE* (Vol. 4, pp. 1942-1948).
- Kloder, S., & Hutchinson, S. (2006, August). Path planning for permutation-invariant multirobot formations. *IEEE Transactions on Robotics*, 22(4), 650-665.
- Kloetzer, M., & Belta, C. (2007, April). Temporal logic planning and control of robotic swarms by hierarchical abstractions. *IEEE Transactions on Robotics*, 23(2), 320 - 330.
- Lalish, E., Morgansen, K., & Tsukamaki, T. (2006). *Formation tracking control using virtual structures and deconfliction*.
- Langer, D., Rosenblatt, J., & Hebert, M. (1994, December). A behavior-based system for off-road navigation. *IEEE Transactions on Robotics and Automation*, 10(6), 776-783.
- Lawton, J., Beard, R., & Young, B. (2003, December). A decentralized approach to formation maneuvers. *IEEE Transactions on Robotics and Automation*, 19(6), 933-941.
- Milutinovi, D., & Lima, P. (2006, December). Modeling and optimal centralized control of a large-size robotic population. *IEEE Transactions on Robotics*, 22(6), 1280 - 1285.
- Mourikis, A., & Roumeliotis, S. (2006a, october). Optimal sensor scheduling for resource-constrained localization of mobile robot formations. *IEEE Transactions on Robotics*, 22(5), 917 - 931.
- Mourikis, A., & Roumeliotis, S. (2006b, August). Performance analysis of multirobot cooperative localization. *IEEE Transactions on Robotics*, 22(4), 666 - 681.
- Pallottino, L., Scordio, V., Bicchi, A., & Frazzoli, E. (2007, December). Decentralized cooperative policy for conflict resolution in multivehicle systems. *IEEE Transactions on Robotics*, 23(6), 1170 - 1183.

- Parker, L. (1998, April). Alliance: an architecture for fault tolerant multirobot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2), 220–240.
- Peasgood, M., Clark, C., & McPhee, J. (2008, April). A complete and scalable strategy for coordinating multiple robots within roadmaps. *IEEE Transactions on Robotics*, 24(2), 283 - 292.
- Ren, W., & Beard, R. (2003, June). *A decentralized scheme for spacecraft formation flying via the virtual structure approach*. Saska, M., Hess, M., & Schilling, K. (2007, December). *Path planning and motion coordination for compact vehicle-formations*. Guimaraes, Portugal.
- Saska, M., Hess, M., & Schilling, K. (2008, May). *Efficient airport snow shoveling by applying autonomous multi-vehicle formations*. Pasadena, USA.
- Saska, M., Macas, M., Preucil, L., & Lhotska, L. (2006). Robot path planning using partical swarm optimization of Ferguson splines. *Proc. IEEE/ETFA'06*, 833-839.
- Tanner, H., Loizou, S., & Kyriakopoulos, K. (2003). Nonholonomic navigation and control of cooperating mobile manipulators. *IEEE Transactions on Robotics and Automation*., 19(1), 53-64.
- Vig, L., & Adams, J. (2006, August). Multi-robot coalition formation. *IEEE Transactions on Robotics*, 22(4).
- Vose, M. D. (1999). *The simple genetic algorithm: Foundations and theory*. MIT Press, Cambridge.
- Ye, J., & Qu, R. (1999). *Fairing of parametric cubic splines*. Elseviers.

IntechOpen



Particle Swarm Optimization

Edited by Aleksandar Lazinica

ISBN 978-953-7619-48-0

Hard cover, 476 pages

Publisher InTech

Published online 01, January, 2009

Published in print edition January, 2009

Particle swarm optimization (PSO) is a population based stochastic optimization technique influenced by the social behavior of bird flocking or fish schooling. PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. This book represents the contributions of the top researchers in this field and will serve as a valuable tool for professionals in this interdisciplinary field.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Martin Macas, Martin Saska, Lenka Lhotska, Libor Preucil and Klaus Schilling (2009). Path Planning for Formations of Mobile Robots using PSO Technique, Particle Swarm Optimization, Aleksandar Lazinica (Ed.), ISBN: 978-953-7619-48-0, InTech, Available from:
http://www.intechopen.com/books/particle_swarm_optimization/path_planning_for_formation_of_mobile_robots_using_pso_technique

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2009 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen