Route Scheduling Approach for Airport Snow Shoveling using Formations of Autonomous Ploughs

Martin Saska, Martin Hess and Klaus Schilling
Julius-Maximilians University of Wuerzburg
Wuerzburg, Germany
{saska,m.hess,schi}@informatik.uni-wuerzburg.de

***Abstract*—This paper presents a route scheduling algorithm for efficient snow shoveling of big international airports. The keynote of the project consists in using temporary formations of autonomous snowploughs. The size of the formations depends on the width of runway to shovel, which supposes splitting and merging of the teams during the cleaning process.**

The route scheduling method itself is based on the design of cleaning coalitions which are adapted for each specific task. The decisions about the optimal team composition and the optimal sequence of the tasks are taken by an heuristic approach that is combined with an exploration of the space of solutions on a limited horizon.

Furthermore in the paper a formation driving approach is discussed. The method commands the robots to form a formation which optimally covers the roads surface and enables to push the snow by their shovels to the sides effectively. The scheduling method as well as the formation driving approach were verified by simulations and by hardware experiments.

***Index Terms*—Mobile robotics, Route Scheduling, Formations driving, Path planning**

I. Introduction

To maintain airport operations safe and uninterrupted is the main task of the ground staff. One of the critical period for the safeness of air traffic is snowy weather. This is not because of reduced visibility during landing, but mainly due to runway conditions. It is a complicated task to remove the snow from the huge surface of the main runways and the big amount of auxiliary roads that are necessary for the planes as well as for the ground vehicles.

Today the tracks of airports are freed from snow by utilizing a fleet of human driven snowploughs. An increase of efficiency, but also a saving of expenses (now the crew of ploughs must be on the alert day and night all winter) could be achieved using an autonomous system of mobile robots. Due to the fact that the big airports are already equipped by the essential sensors, i.e. a global positioning system and automatic detection of runway conditions, such system can be set up relatively easily. Also the periodicity of the task and absence of the obstacles during the cleaning process (e.g. in comparison to highways) predestinate the use of autonomous robots.

A description of an appropriate autonomous system is provided in this paper. The basic idea is motivated by current approaches commonly used for shoveling of runways by human driven snowploughs. The main runway should be cleaned up at once by a sufficiently big group of vehicles, which enables to clean this large surface quickly and a possibility that some snow remains in the landing area is decreased. When the cleaning of the runway is accomplished the big group is splitted to smaller teams with sizes appropriate for shoveling smaller roads. In our case the advantage of such approach could be an opportunity to use a formation driving method that can easily arrange the ploughs to positions optimal for shoveling, but also simplify the robotic system and increase its robustness.

In this paper mainly the problem how to built up the teams of ploughs and how to find an optimal schedule of cleaning tasks will be discussed. A proposed decision-making process is based on a heuristic approach that is combined with an exploration of the space of solution into limited depth. Such technique is able to find a sufficient solution in real time and in addition it can easily respond to dynamic changes in the environment.

The snow shoveling task addressed in this paper is related to the field of cooperative sweeping strategies, whose aim is to find and execute a motion for the robots in order to cover a predefined area by their effecters, which are in our case the snowploughs' shovels. Often it is desired to coordinate the vehicles in a way that time optimality is achieved. Kurabayashi et al. addressed the problem of cooperative sweeping by generating a path for a single robot which is then segmented and distributed among the vehicles [7], [6]. A decentralized approach using an on-line negotiation mechanism to resolve the task sharing was proposed in [10]. Luo et al. developed a real-time method based on biologically inspired neural networks where each robot treats the other robots as moving obstacles [8], [9]. Projecting the task of cooperative sweeping onto the airfield snow shoveling generates some problems that have not yet been addressed. Mainly the planning task varies from the methods mentioned above. Ploughs have to cover more structured environment due to complicated map of airports with amount of various runways irregularly crossing each other. In our problem there is also stronger connection between the robots during the cleaning task. Only one plough usually cannot clean the road alone, hence a bigger cooperative team is necessary.

The rest of the paper is organized as followed: Section II, which is the core of the paper, is composed from two main parts. While in subsection II-A a complexity study should be a motivation for using the proposed scheduling method, the second subsection II-B describes details of such approach. In the section III a formation driving algorithm is discussed followed by results of experiments presented in section IV. Finally conclusions and possible directions of future work are mentioned in section V.

II. Route scheduling approach

In this section the method for route scheduling will be introduced. The main task of the algorithm is to prepare a sorted list of roads that should be visited by the robots. In the list has to be included exact time when the relevant road must be shoveled as well as an information about the robots cleaning the road. This could prevent possible collisions and enable to cover the road in an optimal way.

High complexity of the airport together with indispensable number of ploughs makes this combinatorial problem difficult. In the following subsection will be shown by a complexity study that an optimal solution cannot be found in real time.

A. Complexity study

A natural description of the complicated airport structure could be a graph. All runways as well as smaller roads can be described by multiple edges where the multiplicity is equal to the number of ploughs needed for their shoveling. Similarly crossings of the roads or places where the runway is changing to a narrower path can be replaced by nodes in the graph.

In such well defined structure an optimal coverage by multiple robots can be found using a graph method. The optimal coverage means that each simple edge is visited by a robot at least one time and total time needed for the complete cleaning is minimal. Let's analyze using of the simple "breath first" algorithm where can be easily proven an ability to find the global optimal solution.

"Fig. 1" shows a scenario with two runways a, e connected by three smaller roads b, c, d. For simplification is postulated that the field is composed from two identical squares and time needed for shoveling each side is identical and does not depend on the cleaning track. An extension caused by different radii of the curves as well as time lost by turning on the spot are in this case vanished. It means that e.g. the way from node 3 to 4 along the big runway is in both tracks always two times longer than the shortest way from 1 to 3.

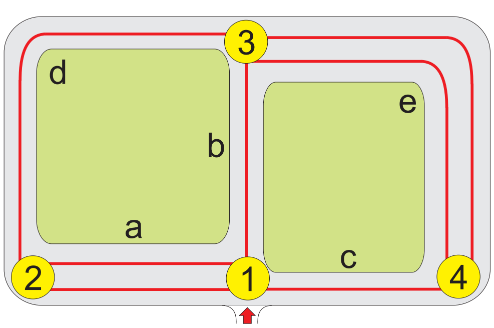


Fig. 1. Shoveling scenario containing 4 nodes, 2 bigger roads and 3 auxiliary roads.

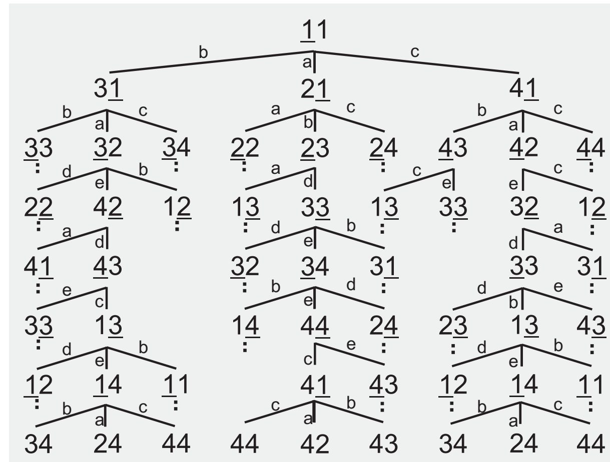


Fig. 2. Reduced tree with an optimal schedule in the middle column.

Let's analyze the situation where two robots are coming to node 1 facing the arrow in a short interval. This state is labelled by vector $\underline{1}1$ in the root of the tree depicted in "Fig. 2". The first underlined digit represents the robot that is already prepared in the crossroad 1 while the second component of the vector without underline means that the second robot is still coming to the node. On the assumption that both robots can shovel the airport independently, the first robot can immediately continue to one of the following nodes 2, 3, 4 using roads a, b, c. The new possible states created by expanding of the root are depicted in the second level of the tree. Similarly level 3 is achieved using the same roads by the second plough. In our simple case the complete tree of depth equal to 8 must be explored to find the optimal solution. In the picture of the tree most of branches are replaced by dots due to size of figure, because the complete tree contains about 1000 leaves. The path from the root to a leaf which corresponds to the complete cleaning sequence with minimal total time will be the desired solution. In the "Fig. 2" an optimal solution can be found in the middle column of the tree, but as well as in the middle columns of the left branch or the right branch.

It is obvious that a tree with all leaves, whose total cleaning time of corresponding solution is less than or equal to the time of the best solution, has to be explored completely to prove optimality of the found schedule. Minimal depth of the tree then will be equal to the number of tracks on condition that there exists a solution with cleaning time equal to the sum of the times needed for cleaning all tracks separately. The total number of tracks that need to be cleaned is

$$W = \sum_{e \in E} W_e, \tag{1}$$

where E is set of all edges in the graph and W_e is the number of robots that are necessary for cleaning the edge e. Now we are able to define a simple lower bound of the number of leaves in the tree as

$$n_l = b_{min}^{W}, \tag{2}$$

where b_{min} is the minimal branching factor of the tree. In our simple example

$$n_l = 2^7 = 128, \tag{3}$$

which is much lower than the actual number of leaves in "Fig. 2". Nevertheless for our purpose such lower bound is sufficient. The estimated number of schedules that must be explored for finding an optimal complete coverage of the Frankfurt international airport ("Fig. 3"), which was chosen as an example for verification of results, is

$$n_l = 3^{761}. \tag{4}$$

The number of tracks that should be cleaned by the robots was obtained taking width of all roads from the airport map and width of the runway sweeper RS 200 [5] usually used for cleaning of airports. It is obvious that the complete tree with number of leaves equal to 10^{363} cannot be created.

An idea to reduce the above mentioned problem is to consider several tracks covering each road as one edge. This means that shovelling of each road will be described as one task[1] that must be accomplished by a sufficient number of robots at one time. Such simplification makes sense, because each runway should be shovelled at once. The advantages of such approach were discussed in section I. The minimal number of ploughs used for the scenario of Frankfurt airport is then equal to 17, because the width of the biggest runway is about 60 meters and working width of the sweepers is 3.6 meters.

Using this restriction the exponent in the equation 4 will be equal to the number of the tasks and therefore significantly lower. Nevertheless the estimated number of successors of each state (the branching factor of the tree) has to be changed. In the previous case this number was equal to the number of roads adjacent to the crossroads, because only one robot could be in the crossing at the same time. Now in the same crossroad can be up to 17 robots (e.g. after cleaning a main runway which needs all ploughs) and the successors must cover all possible options how the robots can continue. The total number of successors can be computed by

$$S = \sum_{i_1=1}^{n_r+1} \sum_{i_2=1}^{i_1} \dots \sum_{i_{b-2}=1}^{i_{b-3}} i_{b-2}, \tag{5}$$

where b is number of adjacent roads and n_r is number of ploughs. For illustration in the case, where a group of 17 robots comes to a crossing with three adjacent roads, 696 successors should be expanded. This together with still unsatisfactory length of the plan (reduced exponent in the equation 4 is for the Frankfurt airport equal to 147) make the problem still unsolvable in real time.

[1]Each task is defined as a moving of a robot from node to node.

B. Method description

The aim of the presented approach is to reduce both the branching factor of the tree as well as the depth of the tree. The reduction of the final tree that needs to be explored must be so significant that a solution will be found promptly. It should enable real time response to failures of the robots or to changes of the environment like e.g. newly detected obstacles and roadblocks.

1) Heuristics: For the reduction of the big number of successors (equation 5) only the most appropriate branches of the tree should be explored. In the ideal case only the successor belonging to the best solution has to be chosen and the algorithm becomes the "depth-first" search.

In our method we use a simple multi-agent approach for description of the complicated behavior of ploughs. Each cleaning formation is considered to be one object (agent) with an assigned task. When the shovelling of a road is finished the same agent can be used for another road only if the number of ploughs is still sufficient. If for the new task a bigger group is needed the agent has to ask the remaining robots for help. Contrariwise, if the new road is narrower and the group can be reduced, a plough or ploughs can be offered to the other agents as free robots. To assign which road should be shovelled by which agent will be the crucial part of this method.

As mentioned above, it is impossible to prove that a decision will be global optimal without exploring the whole tree of solutions. Nevertheless using an appropriate heuristic the decision could be close to the optimal one. The presented method is composed from two simple steps. In the first a sorted list L_t of all uncleaned roads (tasks) is found. The tasks are arranged in order of priority in which they should be accomplished. In the second step the appropriate and free robots are allocated to the most urgent roads. The easiest but tricky sorting approach is to arrange the roads in order of distance from the beginning of shovelling and from the mean position of the ploughs (both factors should be equally merged). In our case the beginning of shoveling is end of the main runway where the group should be splitted for cleaning the auxiliary roads ("see Fig. 4-5"). This rule guarantees that no road will be omitted during the cleaning process and that the ploughs won't be obliged to move too far for cleaning.

In the allocation part of the method the first N roads from the sorted list of the tasks L_t is taken. The number N has to be maximum integer satisfying inequation

$$\sum_{i=1}^{N} W_i \leq n_{fr}, \tag{6}$$

where W_i is the number of ploughs that are necessary for cleaning of the i^{th} task in the L_t and n_{fr} is the number of free robots[2]. Then the free robots should be assigned to groups F_i so that the term

[2]Free robots can be ploughs of the agent that just finished its task or redundant robots offered by the other agents.

Fig. 3. Picture of the Frankfurt international airport. Dashed rectangle bounds the area that is used in “Fig. 4-6” and in “Fig. 9-10”. Roads with a part belonging to the dotted rectangle have to be cleaned in the final simulation.

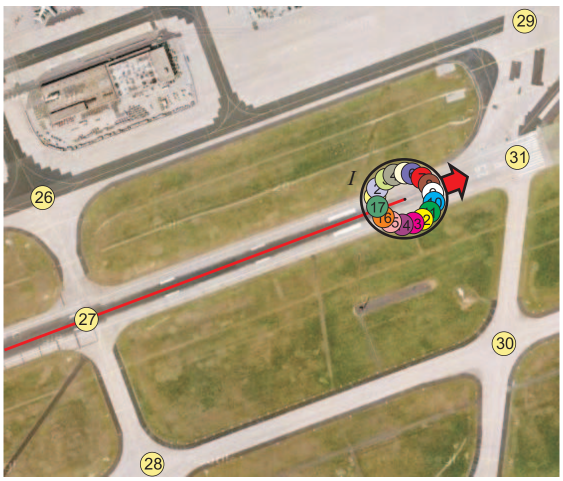


Fig. 4. One big formation is shoveling the part of the main runway between crossroads 27 and 31. Each small bubble denotes one robot. Already cleaned roads are depicted by red line.

$$\max_{i=1..N}\left(\max_{j\in F_i}\left(time_{i,j}\right)\right) \tag{7}$$

is minimal. $time_{i,j}$ is the time needed for the plough j to move from an actual position to the beginning of road i.

Unfortunately the above mentioned heuristics cannot solve all possible situations in a satisfactory manner. An example of obviously wrong decision is shown in “Fig. 4-6”. In the first snapshot of the simulation the big formation of robots I is coming to the end of the main runway and should be splitted for shoveling smaller auxiliary roads. A situation after applying the heuristics is depicted in the second snapshot (“Fig. 5”). While the sub-formation Ia is shoveling the road between crossroads 31 and 30 and the sub-formation Ib is shoveling the road between 31 and 29, the sub-formation Ic is going to clean up road 29-26. In the following snapshot (“Fig. 6”) the formation Ia is cleaning the road between 30 and 28 and the formation Ib is going back to the crossroad 31, which is the shortest way to the road 27-28 that is the next in the sorted list of tasks. It is obvious that the more optimal solution could be to continue with Ib to clean the road 29-26 and to use freed

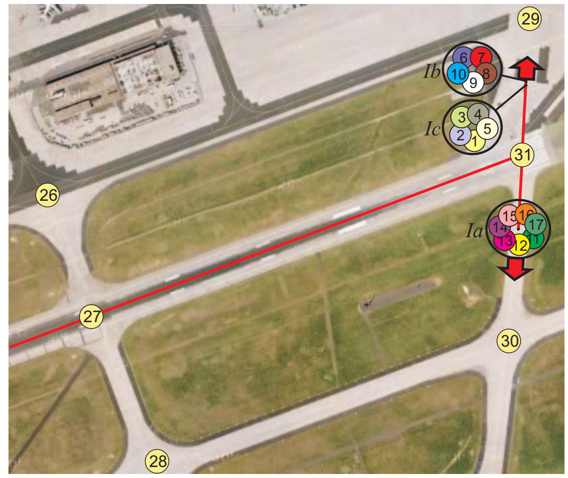


Fig. 5. Formations Ia and Ib are shoveling access roads. Formation Ic is moving to clean road 29-26.

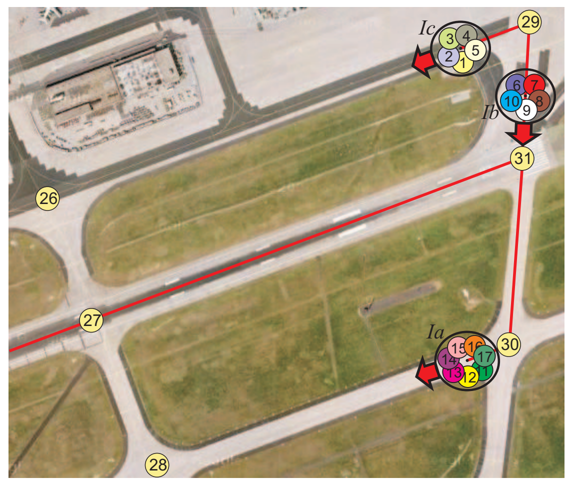


Fig. 6. Formations Ia and Ic are shoveling uncleaned roads 30-28 and 29-26. Formation Ib is going along already cleaned way to shovel road 27-28.

Ic for shoveling another uncleaned way.

Such smarter result should be obtained using more ”sophisticated” heuristics or using an approach, which can enable to expand the local information about the actual state of ploughs. The biggest disadvantage of the heuristics’ extension is impossibility to cover all situations that can happen. The heuristics become too complicated and a debugging of the scheduling process is troublesome. On the contrary the second approach could solve the problem more generally and transparently.

2) Space of solutions exploration: The basic idea is to extend the simple algorithm to an approach valuing a future evolution of several different decisions. The previous method used a fixed set of tasks that should be cleaned by free vehicles in the following step. In some cases such set may not be optimal (an example was shown in “Fig. 4-6”) and it is better to start with different tasks. To find the set of tasks, which can provide a solution with shorter time needed for cleaning all roads of the airport, the following extension was implemented.

In the first step of the method all possible sets of tasks satisfying the inequation 7 are created. Then the sets are arranged in order of sum of tasks' indexes in the sorted list of tasks L_t. Only first b_t sets from such prepared sorted list of sets of tasks L_s will be used for planning (if the length of L_s is less than b_t all sets from L_s will be used). The number b_t will be determined by computation power as an input of the algorithm. As an example that should help to understand the idea of the method let's suppose that each task from the $L_t \in \{A_1, B_2, C_3, D_4\}$ requires exactly one plough and that just two ploughs are prepared for shoveling right now. The list of sets of tasks arranged in order of the sum then will be $L_s \in \{AB_3, AC_4, AD_5, BC_5, BD_6, CD_7\}$. For solutions with the same sum the lowest index in the set is decisive, which again decreases the option that a road will be omitted. It is obvious that the whole tree for all possibilities cannot be explored, but it is not necessary due to the heuristics that prefer the most promising solutions. Notice that in the basic method, described in the subsection II-B1, automatically only the first set AB_3 would be chosen. The idea of the extended method is to try also the other most promising sets and to decide which could be closer to the global optimal solution covering all roads.

In the second step of the extended method only one set of tasks has to be chosen from the first b_t members of L_s. A decision-making process, which is based on partial exploring of the tree of solutions, was developed. The basic idea of the method is described in the "Fig. 7", where the situation depicted in the "Fig. 1" is analyzed. In this example $b_t = 3$ and the depth of explored sub-trees $d_t = 2$.

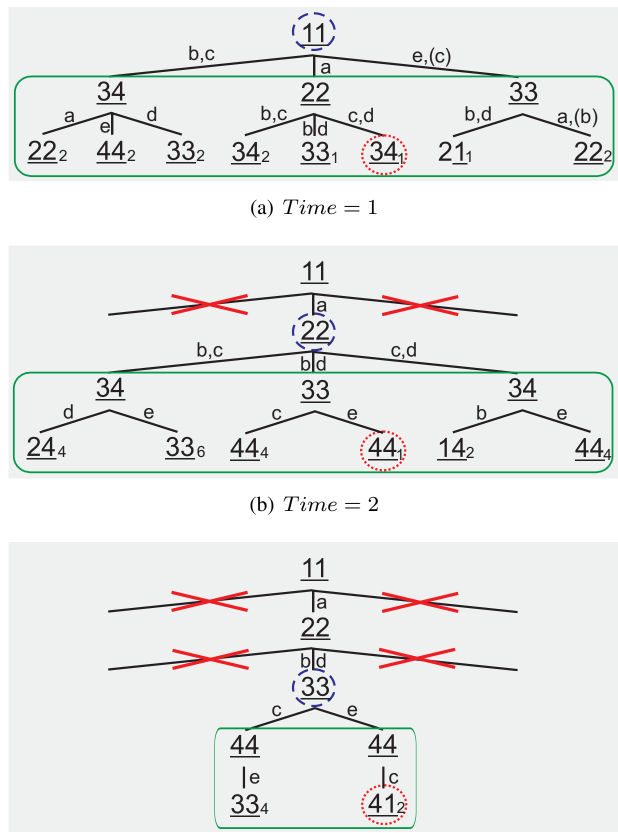
The first tree ("Fig. 7(a)") reflects the beginning of the decision-making process. Both ploughs are in the initial node 1 and the lists necessary for the decision are $L_t \in \{a, b, c, d, e\}$ and $L_s \in \{a, bc, bd, cd, e\}$. The initial state is explored and in the lowest level of the sub-tree a state with the shortest "wasted" time[3] is chosen ($\underline{34}$). On condition that multiple states have the same shortest wasted time a state is chosen randomly. Such states are optimal with respect to the limited information obtained by the partly exploration of the tree. At this moment the ploughs can continue to the next state in the directions of the optimal one. When the new state $\underline{22}$ is reached, the rest of the solution is forgotten and the planning process is restarted from the new state ("Fig. 7(b)"). In the third level of the new sub-tree (fourth level of the tree) the new best state is $\underline{44}$. Such solution may not be global optimal, because for cooperative cleaning of the runway e the plough should be in the same node. Nevertheless the branch of the tree was chosen correctly and in the next step (see "Fig. 7(c)") the schedule can be corrected.

The obtained schedules for the robots $R_1 \in \{1-2-1-3-4\}$ and $R_2 \in \{1-2-3-4-1\}$ are bit longer than schedules found in "Fig. 2", but the restriction that runways have to be cleaned at once was not applied. Another big difference is that there were more than 1000 states evaluated whereas using the extended method only 15. Also the idea of the exploration of states is different. While in the tree "Fig. 2" the successors follow adjacent roads, in the tree in "Fig. 7" the new states are created according to tasks that should be accomplished.

An estimation of the number of the states, that must be evaluated during the complete cleaning, can be calculated due to fixed parameters of the tree quite exactly. Because in each step of the algorithm at least one road should be cleaned, the upper bound of the evaluated states is

$$n = |E|.b_t^{d_t}, \tag{8}$$

where $|E|$ is the number of roads. Parameters of the tree (branching factor b_t and depth d_t) are inputs of the algorithm and should be adjusted according to computing power.



(a) $Time = 1$

(b) $Time = 2$

(c) $Time = 3$

Fig. 7. Illustration of partial construction of the tree in different discrete time. The actual state of the ploughs is marked by a blue dashed circle. New states created from the actual one are within the solid green oval. The best state in the lowest level is marked by a red dotted circle.

III. Technical Details

The main part of this paper is focused on the road scheduling, but even the best schedule is useless without an approach that enables the ploughs to follow the plan. The ploughs should be able to form formations appropriate for shoveling, to cover maximal part of runways during their movement as well as to split and reunite temporary groups.

Methods for formation driving are often based on maintaining a certain distance to a moving reference point which

[3]Wasted time is defined as a time that ploughs lose by moving without shoveling or by waiting to complete a formation.

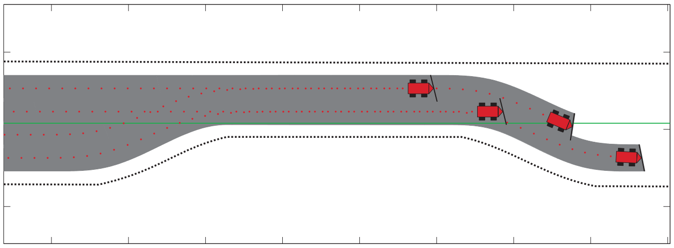


Fig. 8. A formation of 4 snowploughs overcomes a bottleneck along the road by appropriate adapting their spacing.

could be a leading vehicle, the barycenter of the group or even a predefined point. To achieve coordinated driving we rely on the idea to maintain the distance to the reference point in curvilinear coordinates, which is an appropriate way for car-like vehicles with their special restrictions. Using an approach, which was developed within our team, an optimal trajectory for each robot can be easily designed (for method description see [3]).

To follow the desired trajectories a trajectory tracking controller developed for car-like robots has to be implemented on each snowplough. In [1] a controller is described, that utilizes exact feedback linearization on the kinematic equations for the barycenter of the robot located at the center of its rear axle. The bigger the distance between shovel and barycenter of the plough, the more the cleaned part deviates from the desired path. To get around this problem we choose the shovels mount point as the position which should track the desired trajectory. The resulting controller handles sharp turns in a much more efficient way (for details see [11]).

In some cases it is useful to alter the arrangement of a formation while its vehicles are in motion. This can be e.g. in order to adapt to a changing environment like a road getting narrower, to avoid an obstacle, which can be a parking transporter or to adjust the formation structure after switching to a new group. Such deformations are achieved by appropriate and continuous alteration of the size of the formations [11]. An example of such a maneuver can be seen in “Fig. 8”, where a formation consisting of 4 snowploughs overcomes a bottleneck along the road. You can also see that along the whole path the formation should be arranged in such way that the snow is transported to the side of the roads.

IV. Results

A. Simulation

The crucial problem of the presented approach is to find appropriate values of the parameters characterizing the size of the sub-tree that should be explored from an actual state. Results of simulations with different setting of the algorithm are shown in table I. The quality of the cleaning process is represented by the total time needed for complete shoveling of the area situated around the first runway of the Frankfurt international airport (“Fig. 3”). It is logical that better solutions are found with increasing depth of the explored sub-tree d_t, which corresponds to the horizon of planning, as well as with spreading of the branching factor b_t. Unfortunately both parameters significantly influence computational time of the algorithm as it is obvious from the equation 8.

TABLE I

Total time of schedules found with algorithm using different setting of parameters d_t and b_t

	$d_t = 2$	$d_t = 3$	$d_t = 4$	$d_t = 7$
$b_t = 2$	2026	1930	1712	1712
$b_t = 3$	2026	1779	1539	1539
$b_t = 4$	1802	1773	1539	1539
$b_t = 7$	1802	1539	1539	1539

The best solution is found by setting the parameters $b_t \geq 3$ and $d_t \geq 4$. The number of states that were evaluated always when a new agent was created or when new tasks were distributed among existing agents was in the best case only 81. This enables to find a decision in fractions of milliseconds and decision-making process can quickly responds to sudden changes in the workspace.

Snapshots of the simulation using the best solution are depicted in figure 9 and 10. The first picture shows a situation after splitting the main formation. In contrast to the picture 5 the robots are distributed optimally and the redundant movement of two formations along the road 31-29 is corrected. The complete animation of the shoveling process can be found on the web-site [4].

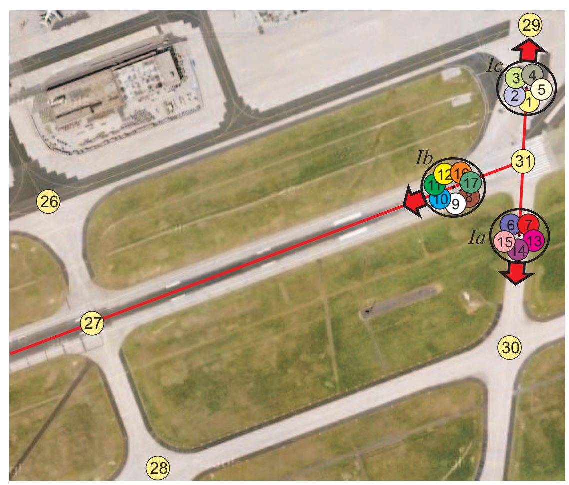


Fig. 9. Corrected situation from figure 5. Formations Ia and Ic are shoveling roads 31-30 and 31-29. Formation Ib is going back to shovel road 27-26.

B. Hardware experiment

In the presented simulations the movement of the ploughs and their arrangement in the cleaning formations where simplified, because this paper is focused mainly on the optimal tasks distribution. Nevertheless for understanding the whole idea it is important to show at least basic abilities of the robots in real experiment. For this purpose we used MERLIN platform of the University of Wuerzburg [12]. The MERLIN-Testbed consists of multiple homogeneous car-like robots which are equipped with various sensors like wheel encoders, gyroscope

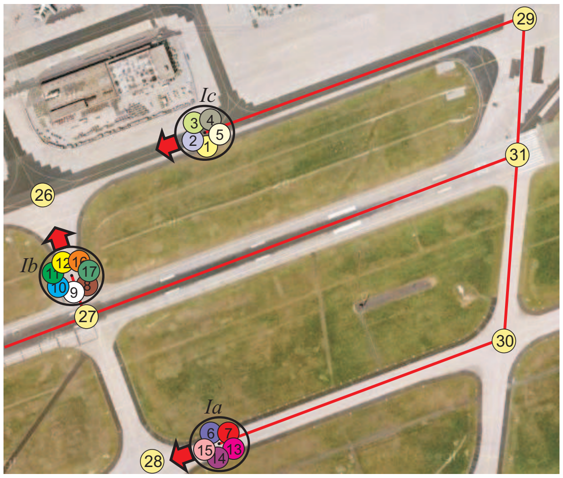


Fig. 10. Three formations with variable number of ploughs are shoveling part of international airport in Frankfurt.

and ultrasonic range finders. Each vehicle also contains a wireless communication device to enable inter-robot data-exchange.

For the experiments the snow was made of small pieces of polystyrene and the shovels are simply straight bars, which were mounted transversely to the bumper of the mobile robots (details are described in [2]). The scenario includes the most important skills as is shoveling in the formation, splitting of the group and finally reuniting back to formation. Pictures of the hardware experiment presented in the "Fig. 11" could be a motivation to download complete movie which is also available on the web-site [4].

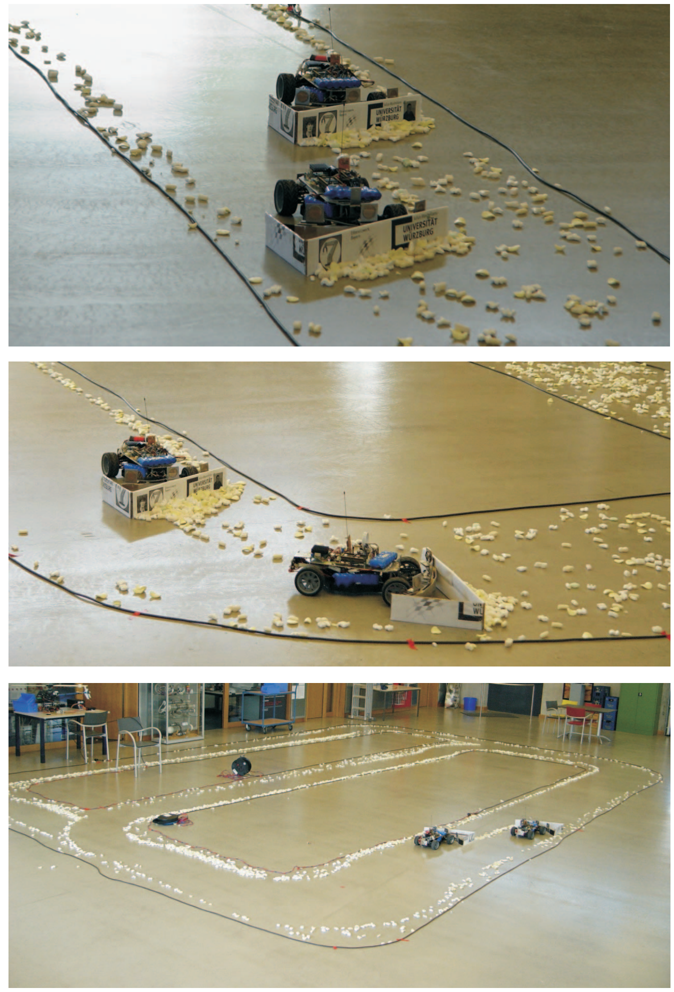


Fig. 11. Snapshots of the hardware experiment. Robots are cooperatively shoveling a big runway which is followed by splitting and cleaning narrower roads independently.

V. Conclusions and Future Works

A. Conclusions

In this paper a novel route scheduling approach for autonomous airport snow shoveling was introduced. The proposed method is based on building temporary coalitions of mobile robots which can be splitted or merged depending on variable width of the runways. In the decision-making process a heuristic approach is combined with an exploration of the space of solutions into limited depth. Such technique is able to find a sufficient solution in real time and it can easily respond to dynamic changes in the environment.

The robustness of the method was increased using a formation driving algorithm which enables to avoid collisions within the group of ploughs shoveling the same road. Using this method the robots are arranged to push the snow by their shovels to the sides and to cover the road optimally. While the functionality of the scheduling approach was verified by various simulations, the formation driving algorithm was tested in hardware experiments.

B. Future Works

The presented project should continue with a more complicated hardware experiment. We would like to prepare an experiment in a more realistic environment with moving obstacles and with a structure comparable to the map of a real airport. This can only be realized by an outdoor experiment which supposes to extend the algorithm for outdoor robots. Such schema should examine the abilities of the scheduling method not only the driving skills that have been proven already. Mainly the ability to respond to sudden changes in the structure of the map (e.g. temporary closed roads) or to replace a failed plough should be tested.

Acknowledgment

This work was supported by the Elitenetwork of Bavaria through the program "Identification, Optimization and Control with Applications in Modern Technologies".

References

[1] A. De Luca, G. Oriolo, and C. Samson, "Feedback control of a nonholonomic car-like robot," in *Planning robot motion*, J.-P. Laumond, Ed. Springer-Verlag, 1998.

[2] M. Hess, M. Saska, and K. Schilling, "Autonomous multi-vehicle formations for cooperative airfield snow shoveling," in *Proc. of the 3rd European Conference on Mobile Robots (ECMR07)*, Freiburg, Germany, 2007.

[3] ——, "Enhanced motion planning for dynamic formations of nonholonomic mobile robots," in *Proc. of the 6th IFAC Symposium on Intelligent Autonomous Vehicles (IAV2007)*, September 2007.

[4] http://www2.am.uni-erlangen.de/elitenetzwerk-optimierung, *[cit. 2008.3.10]*, International Doctorate Program within the Elite Network of Bavaria. Link: movie.

[5] http://www.overaasen.no/runway-sweepers, *[cit. 2008.3.10]*, Runway Sweepers from Overaasen Snow Removal Systems.

[6] D. Kurabayashi, J. Ota, T. Arai, S. Ichikawa, S. Koga, H. Asama, and I. Endo, "Cooperative sweeping by multiple mobile robots with relocating portable obstacles," in *Proc. of the Conf. on Intelligent Robots and Systems*, Osaka, Japan, 1996, pp. 1472 – 1477.

[7] D. Kurabayashi, J. Ota, T. Arai, and E. Yoshida, "Cooperative sweeping by multiple mobile robots," in *Proc. of the Conf. on Robotics and Automation*, Minneapolis, Minnesota, 1996.

[8] C. Luo and X. Yang, "A real-time cooperative sweeping strategy for multiple cleaning robots," in *Proc. of the Symp. on Intelligent Control*, Vancouver, Canada, 2002.

[9] C. Luo, X. Yang, and D. Stacey, "Real-time path planning with deadlock avoidance of multiple cleaning robots," in *Proc. of the Conf. on Robotics and Automation*, Taipei, Taiwan, 2003.

[10] T. Min and H. Yin, "A decentralized approach for cooperative sweeping by multiple," in *Proc. of the Conf. on Intelligent Robots and Systems*, Victoria, Canada, 1998.

[11] M. Saska, M. Hess, and K. Schilling, "Efficient airport snow shoveling by applying autonomous multi-vehicle formations," in *Proc. of IEEE International Conference on Robotics and Automation (ICRA 2008)*, Pasadena, California, 2008.

[12] K. Schilling and Q. Meng, "The merlin vehicles for outdoor applications," in *Unmaned Ground Vehicle Technology IV, Proceedings of SPIE*, G. R. Gerhart, C. M. Shoemaker, and D. W. Gage, Eds., vol. 4715, 2002, pp. 43–49.