Task sequencing for autonomous robotic vacuum cleaners

Cite as: AIP Conference Proceedings **1863**, 560044 (2017); https://doi.org/10.1063/1.4992727
Published Online: 21 July 2017

Anna Gorbenko and Vladimir Popov

Task Sequencing for Autonomous Robotic Vacuum Cleaners

Anna Gorbenko and Vladimir Popov

Ural Federal University, Department of Intelligent Systems and Robotics, Mathematics and Computer Science Institute, Lenin st., 51, 620083 Ekaterinburg, Russian Federation

Abstract. Various planning problems for robotic systems are of considerable interest. One of such problems is the problem of task sequencing. In this paper, we consider the problem of task sequencing for autonomous vacuum floor cleaning robots. We consider a graph model for the problem. We propose an efficient approach to solve the problem. In particular, we use an explicit reduction from the decision version of the problem to the satisfiability problem. We present the results of computational experiments for different satisfiability algorithms.

Keywords: task sequencing; robotic vacuum cleaners; **NP**-complete; satisfiability problem; satisfiability algorithms
PACS: 07.05.Mh

INTRODUCTION

Autonomous service robots have been getting popular in recent years (see e.g. [1, 2]). There are a number of possible applications of service robots. In particular, we can mention industrial service robots, domestic robots, and scientific robotic systems. Autonomous service robots are used extensively for the solution of various tasks. They can clean floors, mow lawns, guard homes, do some surgeries, inspect pipes and sites that are hazardous to people, fight fires, and defuse bombs. This paper will focus on robotic vacuum cleaners (see e.g. [3, 4, 5, 6]).

Different planning problems of robotic systems received a lot of attention recently (see e.g. [7, 8, 9]). In this paper, we consider the problem of task sequencing for autonomous robotic vacuum cleaners.

PRELIMINARIES AND PROBLEM DEFINITION

There are a number of different approaches that can be used for the solution of the problem of task sequencing for autonomous robotic vacuum cleaners (see e.g. [10, 11, 12, 13]). It is clear that we can consider floors of indoor environments as grid graphs (see e.g. [14]). However, if we consider natural indoor environments, then we need to take into account a number of various factors that have significant influence on the level of terrain traversability. In particular, some cells may contain various untraversable segments; the presence of different wires considerably reduces the traversability and adds some restrictions on the direction of motion; for some cells, there is possibility of systematic appearances of movable obstacles (chairs, toys, vases and so on); different cells have different probability of appearances of various mobile obstacles (people, pets, robots). Moreover, the motion of the robot can cause systematic changes of the environment. Many of such factors can be naturally formalized as restrictions on the direction of motion. So, we consider maps of indoor environments as directed graphs (see e.g. [15]).

Now we define the formal model of indoor environment floor cleaning. We consider a directed graph as an ordered pair $G = (V,A)$ where V is a set of nodes and A is a set of ordered pairs of nodes, called arcs. In general, robotic vacuum cleaners have a number of charging stations. The robotic cleaner automatically recharges its power. Frequently, it is assumed that the robotic cleaner will automatically find the charging station to recharge when it starts to run low on power. However, the robot can use various charging policies. Let V_C be the set of nodes that represent locations of charging stations. A cleaning robot is able to function in autonomous mode excepting the case of cleaning a dust collector. In the case of dust collector overfull the robot stops working, through sound signal indicates its state, and requires emptying dust collector. Let V_E be the set of nodes that represent locations that can be used for emptying dust collectors. We assume that G is a map of some indoor environment if and only if

- $V = U \cup W \cup C \cup E$;
- sets U, W, C, and E are disjoint;

International Conference of Numerical Analysis and Applied Mathematics (ICNAAM 2016)
AIP Conf. Proc. 1863, 560044-1–560044-4; doi: 10.1063/1.4992727
Published by AIP Publishing. 978-0-7354-1538-6/$30.00

- $U = \{u_1, \ldots, u_n\}$, $W = \{w_1, \ldots, w_n\}$, $C = \{c_{i_1}, \ldots, c_{i_m}\}$, $E = \{e_{j_1}, \ldots, e_{j_k}\}$;
- $(u_i, u_j) \in A \Leftrightarrow (w_i, w_j) \in A \Leftrightarrow (w_i, u_j) \in A \Leftrightarrow (u_i, w_j) \in A$;
- $1 \le i_1 < \ldots < i_m \le n$, $1 \le j_1 < \ldots < j_k \le n$;
- $M = \{i_1, \ldots, i_m\}$, $K = \{j_1, \ldots, j_k\}$;
- $(u_i, c_i), (c_i, u_i), (w_i, c_i), (c_i, w_i) \in A$, $i \in M$;
- $(u_i, e_i), (e_i, u_i), (w_i, e_i), (e_i, w_i) \in A$, $i \in K$;
- $(c_i, e_i), (e_i, c_i) \in A$, $i \in K \cap M$;
- $(u_i, u_j) \in A \Leftrightarrow (u_i, c_j) \in A \Leftrightarrow (w_i, c_j) \in A$, $j \in M$;
- $(u_i, u_j) \in A \Leftrightarrow (c_i, u_j) \in A \Leftrightarrow (c_i, w_j) \in A$, $i \in M$;
- $(u_i, u_j) \in A \Leftrightarrow (c_i, c_j) \in A$, $i, j \in M$;
- $(u_i, u_j) \in A \Leftrightarrow (u_i, e_j) \in A \Leftrightarrow (w_i, e_j) \in A$, $j \in K$;
- $(u_i, u_j) \in A \Leftrightarrow (e_i, u_j) \in A \Leftrightarrow (e_i, w_j) \in A$, $i \in K$;
- $(u_i, u_j) \in A \Leftrightarrow (e_i, e_j) \in A$, $i, j \in K$;
- $(u_i, u_j) \in A \Leftrightarrow (c_i, e_j) \in A$, $i \in M$, $j \in K$;
- $(u_i, u_j) \in A \Leftrightarrow (e_i, c_j) \in A$, $j \in M$, $i \in K$.

We assume that nodes u_i, w_i, c_i, and e_i represent the same part of the environment, $1 \le i \le n$. If a path contains u_i, then the robot cleans the part of the environment that represented by u_i, $1 \le i \le n$. If a path contains w_i, then the part of the environment that represented by w_i used for transition, $1 \le i \le n$.

Let $f_S(x) \in \{1, 0\}$, $x \in A$. We assume that $f_S(x) = 1$ if and only if $x \in S$. For any path $x_1, x_2, \ldots, x_r$, let $f_S(x_1, x_2, \ldots, x_r) = f_S(x_1, x_2, \ldots, x_{r-1}) + f_S(x_r)$ where $r > 1$, $x_1, x_2, \ldots, x_r \in V$.

THE PROBLEM OF TASK SEQUENCING FOR AUTONOMOUS ROBOTIC VACUUM CLEANERS (TS):

INSTANCE: *Map $G = (V, A)$ of indoor environment, positive integers R, Num_C, Num_E, Max_E.*

QUESTION: *Is there a path $x_1, x_2, \ldots, x_r$, $r \le R$, such that*

- $f_C(x_i, x_{i+1}, \ldots, x_{i+Num_C}) > 0$, $1 \le i \le r - Num_C$;
- *if* $f_U(x_i, x_{i+1}, \ldots, x_{i+p}) > Num_E$ *where* $1 \le i < i + p \le r$, *then* $f_E(x_i, x_{i+1}, \ldots, x_{i+p}) > 0$;
- $f_E(x_1, x_2, \ldots, x_r) < Max_E$;
- $f_{\{u_i\}}(x_1, x_2, \ldots, x_r) = 1$, $1 \le i \le n$.

We need conditions $f_U(x_i, x_{i+1}, \ldots, x_{i+p}) > Num_E \Rightarrow f_E(x_i, x_{i+1}, \ldots, x_{i+p}) > 0$ to avoid dust collector overfull. We use the condition $f_E(x_1, x_2, \ldots, x_r) < Max_E$ to minimize human assistance. Conditions $f_{\{u_i\}}(x_1, x_2, \ldots, x_r) = 1$, $1 \le i \le r$ guarantee us the fullness of cleaning and lack of excess work. We need conditions $f_C(x_i, x_{i+1}, \ldots, x_{i+Num_C}) > 0$ to avoid battery discharges.

AN EXPLICIT REDUCTION FROM TS TO THE SATISFIABILITY PROBLEM

Let $R = n$, $Max_E = n + 1$, $Num_E = n$, and $Num_C = n$. Let $C = E = \emptyset$. Also, we assume that $(u_i, u_j) \in A$ if and only if $(u_j, u_i) \in A$. If we consider the problem TS under such restrictions, then it is easy to see that we obtain the Hamiltonian path problem (see e.g. [16]). It is well known that the Hamiltonian path problem is **NP**-complete ([16], theorem 9.7). So, the problem TS is **NP**-hard.

The 3-satisfiability problem (3SAT) is one of the most well studied **NP**-complete problems. Encoding different hard problems as instances of 3SAT has recently caused considerable interest (see e.g. [17, 18]). Note that 3SAT is the problem of determining if the variables of a given boolean function in conjunctive normal form with 3 variables per clause (3-CNF) can be assigned in such a way as to make the formula evaluate to true (see e.g. [19]). In this paper, we consider an explicit reduction from TS to 3SAT and try to solve TS with satisfiability algorithms.

Let

$$\wedge_{1 \le i \le R} \vee_{1 \le j \le 4} x[i, j], \tag{1}$$

$$\wedge_{1 \le i \le R, 1 \le j[1] < j[2] \le 4} (\neg x[i, j[1]] \vee \neg x[i, j[2]]), \tag{2}$$

$$\wedge_{1 \le i \le R, 1 \le j \le 4} \neg x[i, j] \vee (\vee_{1 \le l \le L_j} y[i, j, l]), \tag{3}$$

$$\wedge_{1 \le i \le R, 1 \le j \le 4, 1 \le l[1] < l[2] \le L_j} (\neg x[i, j] \vee \neg y[i, j, l[1]] \vee \neg y[i, j, l[2]]), \tag{4}$$

TABLE 1. Time of solution for different satisfiability algorithms.

solver data set	A[1] *LW*	A[2] *LW*	A[3] *LW*	A[1] *VC*	A[2] *VC*	A[3] *VC*
average	17.1 sec	14.3 sec	18.5 sec	8.4 sec	10.6 sec	7.3 sec
max	39.2 sec	27.5 sec	38.4 sec	8.9 sec	12.7 sec	11.3 sec
best	8.4 sec	6.2 sec	7.9 sec	7.8 sec	4.6 sec	4.1 sec

$$\wedge_{1\le i\le R-Num_C} \vee_{i\le j\le i+Num_C} x[i,3], \tag{5}$$

$$\wedge_{1\le i\le R, 1\le l\le n}(\neg x[i,1] \vee \neg y[i,1,l]) \rightarrow \neg z[i,l], \tag{6}$$

$$\wedge_{1\le l\le n} \vee_{1\le i\le R} z[i,l], \tag{7}$$

$$\wedge_{1\le j\le R, 1\le i\le n} s[i,j] \rightarrow (\vee_{1\le l\le n} z[j,l]), \tag{8}$$

$$\wedge_{1\le j[1]\le R, 1\le i[1]\le n, i[2]<i[1], j[2]>j[1]} s[i[1],j[1]] \rightarrow \neg s[i[2],j[2]], \tag{9}$$

$$\wedge_{1\le i\le n} \vee_{1\le j\le R} s[i,j], \tag{10}$$

$$\wedge_{1\le i\le i+Num_E\le n, 1\le j[1]\le R, 1\le j[2]\le R, <i[1], j[2]>j[1]} \neg s[i,j[1]] \vee \neg s[i,j[2]] \vee (\vee_{j[1]\le l\le j[2]} x[l,4]), \tag{11}$$

$$\wedge_{1\le i\le R, 1\le j[2]\le R, <i[1], j[2]>j[1]} \neg x[i,4] \vee \vee(\vee_{1\le j\le Max_E-1} t[i,j]), \tag{12}$$

$$\wedge_{1\le j\le Max_E-1, 1\le i[1]<i[2]\le R}(\neg t[i[1],j] \vee \neg t[i[2],j]), \tag{13}$$

where $L_1 = L_2 = n$, $L_3 = m$, $L_4 = k$. Let ξ be a conjunction of functions (1) – (13). It is easy to see that the function ξ is satisfiable if and only if there is a path $x_1, x_2, \ldots, x_r$, $r \le R$, such that

- $f_C(x_i, x_{i+1}, \ldots, x_{i+Num_C}) > 0$, $1 \le i \le r - Num_C$;
- if $f_U(x_i, x_{i+1}, \ldots, x_{i+p}) > Num_E$ where $1 \le i < i+p \le r$, then $f_E(x_i, x_{i+1}, \ldots, x_{i+p}) > 0$;
- $f_E(x_1, x_2, \ldots, x_r) < Max_E$;
- $f_{\{u_i\}}(x_1, x_2, \ldots, x_r) = 1$, $1 \le i \le n$.

Using standard relations, we can obtain an explicit transformation of the function ξ into a function τ such that $\xi \Leftrightarrow \tau$ and τ is a 3-CNF. It is clear that τ gives us an explicit reduction from TS to 3SAT.

COMPUTATIONAL EXPERIMENTS FOR SATISFIABILITY ALGORITHMS

The robot Neato XV-11 (see e.g. Neato Robotics web page: `http://www.neatorobotics.com/`) is used to perform experiments on the real-world data. To perform computational experiments on the real-world data, we have created the test set *VC* that consists of 100 different maps. We have considered maps of different natural indoor environments. In particular, we have considered various corridor, lab, and flat environments with size from 20 m^2 to 200 m^2. The average size is 60 m^2.

It is easy to see that the problem TS can be naturally used for the solution of the problem of integrated task sequencing and path planning for robotic remote laser welding (see e.g. [20, 21, 22]). In our experiments, we have considered industrial data set *LW* that consists of 40 different stitch layouts for welding. The experiments were performed using a maximum inclination angle of 15° and 30°, which resulted in 80 different instances. The number of stitches in an instance varied between 80 and 120.

In our experiments, we have considered a number of different satisfiability algorithms to obtain optimal solutions of the problem TS. In particular, we have used algorithms A[1] (see [23]), A[2] (see [24]), A[3] (see [25]) for the satisfiability problem. The experiments were run on a Sony Vaio PCG-51111v with Windows 7 Professional. In general case, it should be noted that satisfiability algorithms give us only information about existence of solution. However, each of the algorithms that we have considered trying to obtain the values of the variables for which the boolean function is satisfiable. Therefore, if the boolean function is satisfiable, then we can obtain some assignment for the variables. So, we can obtain a solution for TS. Selected experimental results are given in Table 1.

CONCLUSION

In this paper, we have proposed an approach to efficient optimal solution of of task sequencing for autonomous vacuum floor cleaning robots. Our approach is based on an explicit reduction from the decision version of the problem of task sequencing for autonomous robotic vacuum cleaners to the satisfiability problem. Also, we have considered the problem of task sequencing for autonomous robotic vacuum cleaners for optimal solution of the problem of integrated task sequencing and path planning for robotic remote laser welding.

We have presented the results of computational experiments for different satisfiability algorithms. For industrial data set for robotic remote laser welding, the algorithm A[2] gives us the best results. For indoor environments, the algorithm A[3] gives us the best average time. However, the algorithm A[1] solves each problem of VC in less than 9 seconds. Although we have considered the formulation of the problem of integrated task sequencing and path planning for robotic remote laser welding which is substantially different from the formulation of the authors of [20], it should be noted that solution time for integrated algorithm [20] is 600 seconds. The algorithm A[2] solves each problem of LW in less than 28 seconds.

REFERENCES

1. T. Fong, I. Nourbakhsh, and K. Dautenhahn, *Robot. Auton. Syst.* **42**, 143–166 (2003).
2. J. Kramer, and M. Scheutz, *Auton. Robot.* **22**, 101–132 (2007).
3. J. Fink, V. Bauwens, F. Kaplan, and P. Dillenbourg, *Int. J. Soc. Robot.* **5**, 389–408 (2013).
4. P. Fiorini, and E. Prassler, *Auton. Robot.* **9**, 227–235 (2000).
5. B. Hendriks, B. Meerbeek, S. Boess, S. Pauws, and M. Sonneveld, *Int. J. Soc. Robot.* **3**, 187–195 (2010).
6. F. Vaussard, J. Fink, V. Bauwens, P. Retornaz, D. Hamel, P. Dillenbourg, and F. Mondada, *Robot. Auton. Syst.* **62**, 376–391 (2014).
7. J. Kober, and J. Peters, *Learning motor skills: from algorithms to robot experiments*, vol. 97 of *Springer Tracts in Advanced Robotics*, Springer-Verlag, Cham, 2014, ISBN 978-3-319-03194-1.
8. S. M. LaValle, *Planning Algorithms*, Cambridge University Press, Cambridge, 2006, ISBN 0-521-86205-1.
9. J. Lunenburg, *Context-Aware Design and Motion Planning for Autonomous Service Robots*, Ipskamp Drukkers B.V., Enschede, the Netherlands, 2015, ISBN 978-90-386-3870-6.
10. A. Gorbenko, M. Mornev, and V. Popov, *IAENG IJCS* **38**, 176–182 (2011).
11. A. Gorbenko, and V. Popov, *Journal of Ambient Intelligence and Smart Environments* **7**, 835–847 (2015).
12. V. Popov, *AIP Conference Proceedings* **1558**, 2083–2086 (2013).
13. F. Vaussard, P. Retornaz, D. Hamel, and F. Mondada, “Cutting Down the Energy Consumed by Domestic Robots: Insights from Robotic Vacuum Cleaners,” in *Advances in Autonomous Robotics*, edited by G. H. et al., Lecture Notes in Computer Science 7429, Springer-Verlag, Heidelberg, 2012, pp. 128–139.
14. A. Itai, C. Papadimitriou, and J. Szwarcfite, *SIAM J. Comput.* **11**, 676–686 (1982).
15. R. Diestel, *Graph Theory*, vol. 173 of *Graduate Texts in Mathematics*, Springer-Verlag, Heidelberg, 2005, ISBN 978-3-642-14278-9.
16. C. H. Papadimitriou, *Computational Complexity*, Addison Wesley Longman, Reading, Massachusetts, 1994, ISBN 0-201-53082-1.
17. J. Gu, P. Purdom, J. Franco, and B. Wah, “Algorithms for the Satisfiability (SAT) Problem: A Survey,” in *Cliques, Coloring and Satisfiability: Second DIMACS Implementation Challenge*, edited by D. J. et al., ACM Press, Providence, Rhode Island, 1996, pp. 19–152.
18. Y. Kilani, M. Bsoul, A. Alsarhan, and A. Al-Khasawneh, *IJAIP* **5**, 233–256 (2013).
19. M. Garey, and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, New York, 1990, 2 edn., ISBN 0-716-71045-5.
20. A. Kovács, *International Journal of Production Research* **54**, 1210–1224 (2016).
21. G. Reinhart, U. Munzert, and W. Vogl, *CIRP Annals - Manufacturing Technology* **57**, 37–40 (2008).
22. J. Stemmann, and R. Zunke, *Operations Research Proceedings* **2005**, 729–734 (2005).
23. A. Gorbenko, and V. Popov, *Applied Mechanics and Materials* **365**, 190–193 (2013).
24. V. Popov, *Adv. Studies Theor. Phys.* **7**, 355–358 (2013).
25. A. Gorbenko, *Int. Journal of Math. Analysis* **7**, 2317–2331 (2013).