Scheduling Algorithms Exploring via Robotics Learning

Pavlo Merzlykin[1[0000−0002−0752−411X]], Natalia Kharadzjan[1[0000−0001−9193−755X]], Dmytro Medvedev[1[0000−0002−3747−1717]], Irina Zakarljuka[1], and Liliia Fadeeva[1[0000−0003−4906−2539]]

Kryvyi Rih State Pedagogical University, linuxoid@i.ua, n.a.kharadzjan@gmail.com, dimakrrog@gmail.com, zakarliukairina@gmail.com, fadeyevaliliya@gmail.com

Abstract. The new approach to schedule-related problems learning with use of robotics is reported. The materials are based on the authors' teaching experience within framework of Robotics School at Kryvyi Rih State Pedagogical University. The proposed learning problem may be used both for scheduling algorithms exploring and robotics competitions.

Keywords: Robotics, Scheduling Algorithms, Learning, Line-Following, STEM

1 Introduction

Schedule-related problems appear in every system with limited resources and more than one concurrent candidate for these resources. Such problems are typical for computational systems due to hardware limitations. The most usual examples are processes and threads in multitasking system which are scheduled onto memory, processor time, network etc. The problem of scheduling the motion of hard disk's arm while servicing read and write requests has similar nature. The aim of scheduling algorithm is to assign resources to the activities, which stack their claims for the specific resources, in as effective way as possible while it is impossible to meet all the claims at the same time.

The consideration and examination of these problems and approaches to their solving is a significant aspect of Computer Science professional competencies development. It helps to form the ideas of how does multitasking system work and apply them in own projects. In recent times, robotics have turned out to be a powerful tool for Computer Science learning. This area is becoming cheaper and more accessible even for secondary school students. Nowadays market proposes lots of robotic kits for children of different ages and many robotics competitions are held all around the world.

The competitions challenges [3] often involve engineering problems, floor games, battles, path-finding etc. In this report we come up with the idea of robotics challenge which helps to involve students into schedule-related problems.

2 Results and Discussion

2.1 The Nitty-Gritty of the Problem

The proposed problem is based on classical line-following problems. We have the field with some line tracks and a few coloured “stations” as shown in Fig. 1. The robot represents an unmanned “bus” which shuttles between stations. At random time moments at random stations “passengers” are being spawned. The robot receives this information (the number of passengers and the colour of the station) via infrared sensor. Every passenger has his own destination station. The bus capacity is limited to N passengers. So as soon as it picks some at the station, N is reduced to corresponding number. The aim is to take all the passengers to their destinations for a minimum amount of time.

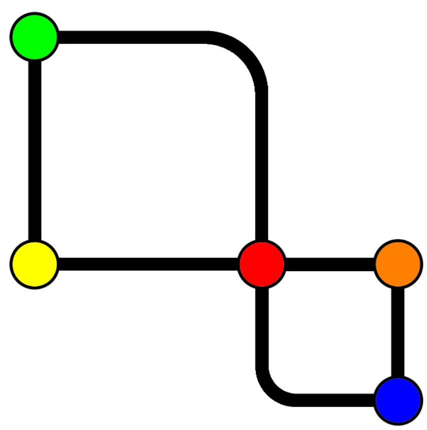

Fig. 1. The example of track

2.2 The Learning Application

This problem is really suitable for discussing scheduling algorithms with students. The limited resource here is the quantity of passengers that the bus can carry at a time. The concurring activities are the passengers waiting at stations.

The simplest approach is First Come First Served (FCFS) [1, 2]. It means that the bus should collect passengers in the order they were reported to appear. The concept of First In First Out (FIFO) queue could be discussed with students within this solution. It is also useful to measure average request processing time.

Greedy algorithms could also be applied. Several problem solving heuristics or greedy strategies are possible here. Thus, robot first collects the passengers from the most “crowded” stations. Another implementation of greedy algorithm is to collect passengers to fill all available places and drive them afterwards to their destinations. It is important to discuss with students the concept of locally optimal solution and test different sets of input values.

The Shortest Job First (SJF), or Shortest Job Next (SJN), approach would mean that at first robot should go to the nearest stations. With proper input data the “starvation” may be simulated here. It means that passengers from far stations should wait a long while. Students could find a solution by using aging technique. As far as we know the time of each request, we can implement the priority system based on the waiting time. Since many aging implementations are possible, students may be asked to discover them on different sets of input data.

The elevator algorithm, which is used for hard drive scheduling, may also easily be simulated in this problem. It would cause the bus to shuttle the same route time after time and collect passengers along the way. Depending on track configuration, many trajectories could be available. So there is the space for another learning research.

Various priority-based schedule schemes could also be simulated in a natural way. Students could implement both fixed and dynamic priority systems in different ways and examine their properties.

For all these algorithms solution time may be measured at the same passengers sequences. It allows to design learning experiments in order to compare algorithms.

2.3 The Robotics Contest Application

A lot of robotics competitions are held all around the world. At the same time, there is no common definite format of such competitions. The most known competition is the World Robot Olympiad (WRO). The Olympiad has been held since 2004 and annually takes place in different countries. The Olympiad is a contest of LEGO robots in three different categories: Regular, Open, Football, and Advanced Robotics Challenge (ARC). For the Regular category, the task is to assemble and program a robot to performs a certain task. The size of the robot is standardized: $25 \times 25 \times 25\ cm$. Participants in the Open category prepare a project for a predefined theme of the season. The tasks for the Regular and Open categories change and become more complicated every year.

Every country and region holds its own competitions, which propose a variety of tasks. However, the most popular ones are the Kegelring (“Bowling Ring”), the passage through labyrinth, the slalom (with a new track every time), the walking robots contest, the biathlon and the competition of robots-sumoists (both autonomous and controlled). The term “autonomous” here means that the robot should be guided only by the program, while “controlled” assumes remote control usage.

Let us consider closer the rules and conditions for the two closest to our problem Olympiad disciplines: walking robots contests and slalom.

The robot (hardware and software) for walking contest is prepared beforehand. The robot must be autonomous and cannot be larger than $25 \times 25 \times 25$ *cm*. During the movement, the robot uses only some points on the surface to support, that is, the robot must move only with help of “legs”. The robot cannot touch the surface by wheels, caterpillars or other revolving parts. Only one microcontroller is allowed in the robot design.

The competition objective is to cross the finish line by all robot’s parts. Duration of each attempt is 2 minutes. Competitions are conducted in two rounds. The first round is qualifying. During the round robots pass through the distance in couples, but time is measured for each robot separately. As a result of the qualifying round, the rating of robots is formed. The referee chooses the number of participants in the final stage. The final stage takes place under the sudden death tournament system. Couples are formed by drawing lots. If the robot does not reach the finish in 2 minutes, it is stopped by the referee.

For walking contest the light-coloured or white field (Fig. 2) is used. The black line width is 2 *cm*. The overall length is usually 236 *cm* and width is 55 *cm* for each robot. The field is skirted with 10 *cm* walls.

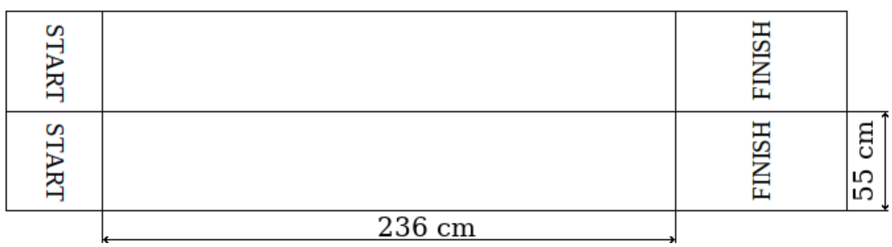

Fig. 2. The track for walking robots competition

Slalom is another competition discipline, which is related to our problem. The competitors should prepare a robot for passing the black line marked trajectory as fast as possible. The robot should be autonomous. Bluetooth ought to be turned off. Robot launching is allowed either by direct program starting by pressing a button on the microprocessor unit or by using the touch sensor. Only standard LEGO parts should be used in solution and the only one microprocessor unit (RCX, NXT, or EV3) may be used in the robot design. Engines and sensors for robots are supplied by LEGO, other manufacturers are prohibited. The use of any non-original or modified (broken, trimmed, curved, etc.) part of LEGO is prohibited. No additional power supplies are allowed. Participants cannot use screws, glues, ropes, rubber, etc. for fastening parts together. It is forbidden to use more than one sensor that detects the line in the one-sensor version of slalom.

The robot's dimensions should not exceed $250 \times 250 \times 250$ mm. Competitions take place on a special limited white field, inside which there is a black 50 mm wide line and elements of the field.

There are several subcategories in the slalom.

1. A high-speed slalom — the robot moves on a given trajectory. The winner is determined by the maximum score and/or by minimum trajectory passing time. The trajectory is represented by a wide line. The minimum radius of curvature is 300 mm. There are no intersections, inversions, breaks, obstacles, and so on. The entire trajectory is in the same plane. To find the elements of a given trajectory, you can use any number of any sensors that are not prohibited by general rules.

2. A high-speed special slalom has the same rules as high-speed slalom, but to find the elements of a given trajectory you can use only one sensor, which is not prohibited by general rules.

The direction of robot's movement may be changed to reverse but not within one qualification phase. The slalom races consist of several qualifying stages. Each qualifying phase consists of at least one check attempt for each team. The maximum number of stages and test attempts is determined by the organizers or, directly, by the referee. Qualifying rounds are conducted between all teams.

During the competition participants may not touch the robot, field and elements of the field on which the competition is being held. At the beginning of the entry it is necessary to delay the robot's movement for 5 seconds. Attempt is considered complete if (i) the robot has finished (it has been registered on the line of the finishing gates), (ii) the robot's projection has gone beyond the motion trajectory line, (iii) the time for the test attempt (180 seconds) has expired, or (iv) the participant has touched the robot, field or field elements. Between the stages it is allowed to change designs and programs of robots, but not during the stage itself.

As we can see, none of reviewed contests provide framework for scheduling algorithms discovering. Most of them assume the navigation in static environment. At the same time, real-world robotics application deals with dynamically changeable environment. So the ability of making effective decisions is crucial for practical purposes. Consequently the scheduling algorithms understanding is important for prospective robotics engineer as well as software engineer. That is why the learning problem we propose is not just mere sport contest. It helps students to develop professional skills.

For the contest purpose, as we concentrate on algorithms, not on design, it is reasonable to use the same robot configuration for all participants. The input request consequences should be also the same during each round, but they may be changed between rounds to provide different types of environments for algorithms testing.

2.4 The Hardware Configuration

The technical requirements for proposed contest include robot with two motors, color sensor, infrared sensor and infrared signal source for sending events. It could be easily assembled with standard LEGO Mindstorms EV3 or similar kit. For example, the small robot model from [4] (fig. 3) would work well.

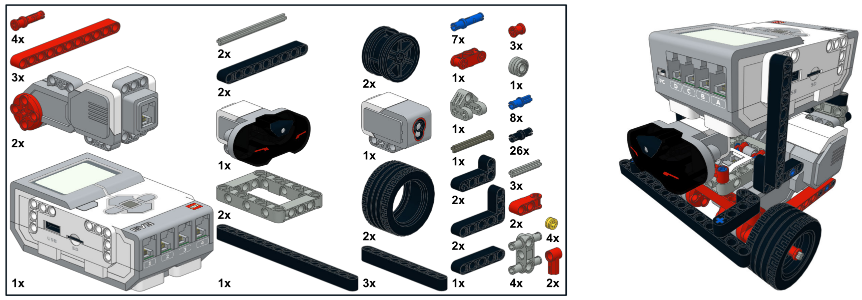

Fig. 3. LEGO Mindstorms EV3-based small robot [4] could be used for proposed problem (needed components set to the left, and assembled model to the right)

For communication purposes both infrared sensor and Bluetooth adapter could be used. EV3 infrared sensor supports four signal channels and allows to distinguish up to 12 IR remote commands (in fact, 11 commands and one void command). It should be enough to implement simple protocol for transmitting resources requests to our robot. The drawback of EV3 infrared sensor is relatively short operating distance (no more than two meters, as we observed, and it really repends on robot's space orientation and environment properties). At the same time, each EV3 block is equipped with Bluetooth adapter. Unlike infrared sensor, it allows robot to send messages as well as receive. Furthermore, it provides more stable and wide-ranged connection. So, while both infrared sensor and Bluetooth adapter are acceptible for our purpose, the latter is preferable.

3 Conclusions

Robotics classes provide excellent educational environment for studying complex Computer Science problems in a natural and effective way via learning experiments. This approach corresponds the concept of STEM education paradigm.

The reported learning problem could be used for scheduling algorithms discovering as well as for robotics competitions. In contrast to the most popular line following problems, which assume the navigation in static environment, our problem is closer to real-world robotics applications and helps students to develop professional skills.

References

1. Nevila Xoxa, Marjo Zotaj, Igli Tafa, Julian Fejzaj Simulation of First Come First Served (FCFS) and Shortest Job First (SJF) Algorithms. IJCSN International Journal of Computer Science and Network, 2014, vol. 3. Available at: http://ijcsn.org/IJCSN-2014/3-6/Simulation-of-First-Come-First-Served-_FCFS_-and-Shortest-Job-First-_SJF_-Algorithms.pdf (accessed 15 January 2018)
2. Zhao, W. and Stankovic, J., A. Performance Analysis of FCFS and Improved FCFS Scheduling Algorithms for Dynamic Real-Time Computer Systems. IEEE 1989, 1989.
3. Dmitry Bazyleva, Alexey Marguna, Konstantin Zimenkoa, Artem Kremleva, Elena Rukujzhaa. Participation in robotics competition as motivation for learning. Procedia - Social and Behavioral Sciences, 2014, pp.835 840. Available at: https://www.sciencedirect.com/science/article/pii/S187704281405397X#fn1 (accessed 15 January 2018)
4. robot-help.ru Small Robot. Available at: https://robot-help.ru/images/lego-mindstorms-ev3/pdf/small-robot-31313.pdf (accessed 15 January 2018)