International Journal of Advanced Computational Engineering and Networking, ISSN(p): 2320-2106, ISSN(e): 2321-2063
Volume-6, Issue-11, Nov.-2018, http://iraj.in

A SYNCHRONOUS MODEL OF SOLVING TRAIN SCHEDULING PROBLEM USING DISTRIBUTED MULTI AGENT SYSTEM

[1]RAZIYEH MOGHADDAS, [2]RULA KHALID, [3]MARWAN ALSHAR'E, [4]SEBASTIN ANTONY JOE

[1,2]Computer Science Department,Gulf College, Oman
Email: [1]raziyeh@gulfcollege.edu.om, [2]rula@gulfcollege.edu.om, [3]marwan@gulfcollege.edu.om, [4]sebastin@gulfcollege.edu.om

Abstract - Train scheduling is an essential problem of railways networks programming. The most common type of railways is single-line which in turn creates more limitations. The efforts to solve this issue have been started since decades, the previous and current efforts work by applying various techniques that contributed to solve the problem in different concepts. However, despite the wide range of multi agent application and the complicated problems that solved by multi agent systems, there were no significant implementation and use of multi agent systems to solve the train scheduling problem. Therefore, we proposed a model for train scheduling in single-line railways, where we present a multi agent based model by formulating this problem as a distributed constraint satisfaction problem (DCSP). Since in railways the trains travel the different block simultaneously, hence the use of intelligent agents and provide a model that can solve the problem simultaneously matches with the issue perfectly. Existing synchronization in our proposed method avoids of unnecessary stop in trains travel and subsequently it gives an optimal scheduling. Experiments designed to evaluate the effectiveness of the suggested solution; the implementation of the algorithm goes on two directions with and without constraints. The result proves the multi agent systems ability to provide a valid contribution to solve the train scheduling issue.

Keywords - Intelligent agents; multi-agent systems; distributed constraint satisfaction problem; train scheduling.

I. INTRODUCTION

Distributed Multi-agent systems are computational systems which several intelligent agents interact together in order to achieve some specific goals. Using agent-based design methods have been one of the most successful solutions in solving lots of complex problems of human processes. For example, a multi-agent systems model to provide support in the disease prevention process [1], using multi-agent approach to scheduling devices in smart homes [2] or using a multi-agent system for monitoring and improving the cloud performance and security [3].

Nowadays, Trains as an important vehicle has a major role in transportation. This role and also particular advantages of trains, compared to other vehicles, has caused much research in relation of improving its efficiency. If the trains do not schedule on a proper plan, it makes the train crash at railroad which such these events cause irreparably cost certainly. To avoid this, finding a scheduling that satisfies the railroad constraints is essential. Accordingly, from many years ago various models have been proposed to solve this problem.

Generally, the train scheduling problem includes determining the arrival and exit times of trains in the stations so that the trains safety and time constraints are satisfied. Considering that most railways in the world are single-line which in turn creates some limitations, so the train scheduling problem is very critical. In these types of lines, only one train can moves between two stations at the same time. While a train is moving in a given path, other trains which want to use this path should wait at the adjacent blocks until the block become available for their move. Trains only can pass along each other in allowed stations. If the train scheduling process is not performed accurately, some accidents and unwanted delays with great damages may be created. Therefore, several models have been proposed since 1973 for train scheduling which can be divided into three groups: mathematical programming, simulation, and modern search methods. In the early 1970 s, Szpiglel [4], have solved the train scheduling problem using a linear programming model. Westerman and Sauder [5] (1983) have presented a decision supporting system for train scheduling in Norfolk Southern Railway Company which led to $ 3 million savings annually. In this system, Brunch and bound technique is used for scheduling with minimum costs. The given system was presented for single rail lines. Mills et al. [6] have modeled the train scheduling problem using network method and non-linear programming technique. Krray [7] has presented the train scheduling problem as a non-linear complex programming model. Generally, solving the mathematical models using the methods such as brunch and bounding is very time consuming. Therefore, the heuristic methods have been proposed to achieve acceptable and optimized results in relatively lower times. Among the heuristic methods, genetic algorithms are more common for solving the train scheduling problems. Tsutsui et al. [8] have studied this solution for the train scheduling problem. Another work was performed by Carey et al. [9] for train scheduling in a railway network with complicated and crowded stations.
In this paper we provide a model for scheduling trains in which several intelligent agents tries to resolve this problem. Multi-agent systems are computational

systems which several agents interact or work together in order to achieve goals. Causes of creation these systems are existence of situations in which a problem needs to be solved in a distributed fashion. For example in situations a central controller is not feasible or because one wants to make good use of the distributed resources. Although, does not much time passes from the creation of such systems, the use of agent-based design methods, has been one of the most successful solutions. The result of this designing method, i.e., a system for problem solving in a distributed mode, is one of the best systems, and is considered as a new tool for solving a variety of human processes.

Distributed constraint satisfaction problem, which presented formally by Sykarv, Yoko and et.al [10], actually, is a distributed form of the standard constraint satisfaction problem. This distributed environment consists of several intelligent agents in which variables are distributed among those agents.All of these agents in trying to achieve a common goalwhich is find an assignment to the variables that satisfy all constraints. This general problem solving is really applicable in many fields of real life problems like resource allocation problems in wireless sensor networks [11], management of urban traffic signals [12], distributed sensor networks [13], and lot of other issues. In other words, any problem that aims is to find a good value allocation for distributed variables can be defined as a DCSP.
It is because that many issues in real world applications can be formulated as a DCSP. Train scheduling is one of these problems. In this paper we present a multi agent based model by considering this problem as a DCSP. In this model each agent owned one of the trains as a variable and controls its constraints. Agents try to resolve the problem simultaneously and independently.

The rest of this paper is organized as follows: Section II an overview of the concepts and discussed topics in this paper. Section III deals to description of our proposed method and its implementation details. Section IV is solving an example and experimental result. Ultimately conclusions are presented in Section V.

II. AN OVERVIEW OF THE CONCEPTS AND DISCUSSED TOPICS

2.1. Definition of Problem

Our proposed model was designed for train scheduling in single-line railways. The general form of this problem is as follows: consider a southern-northern single line railway path. This path includes St stations in its path. The distance between two stations is called a block. Some trains passing these blocks from north to south and vice versa. The origin and destination of each train are known. Also the time required for each train to pass the block is known. The main part of this problem is trains' coincidence limitation. Two trains cannot pass a give path simultaneously. Trains only can pass along each other in allowed stations. The trains' coincidence problem is schematically shown in Fig.1. The x and y axes show time and blocks& stations, respectively. Fig.(a) is a not-allowed coincidence , while Fig.(b) is an allowed coincidence between stations k and k-1.

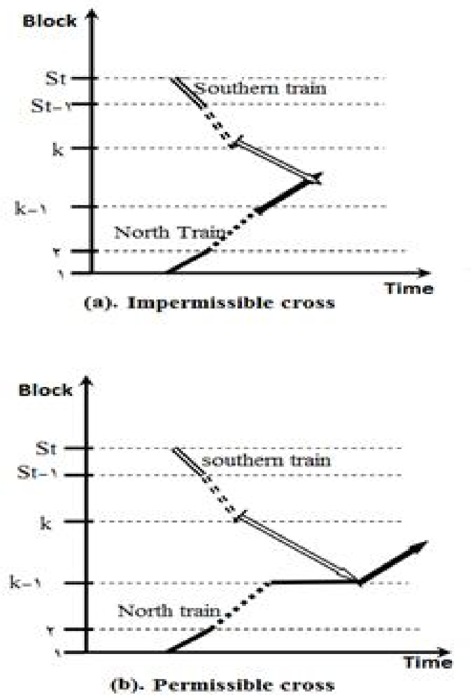

Fig.1. various positions of two trains crossed

2.2. Definition of Agent and Multi-Agent systems

According to [16], an agent is a physical or virtual entity that essentially has the following properties:
1) It is able to live and act in the environment.
2) It is able to sense its local environment.
3) It is driven by certain purposes.
4) It has some reactive behaviors.
However, this is very comprehensive definition, and may be agents have the different definitions for various problems. Multi-agent systems are computational systems which several agents interact or work together in order to achieve goals. In general, two elements should be defined when multi-agent systems are used to solve problems. The first is the meaning and the purpose of each agent and second is actions and behaviors that each agent can take to achieve its purpose. In follows, these elements are described for constraint satisfaction problems.

2.3. Definition of Distributed Constraint Satisfaction Problem

A Distributed Constraint Satisfaction Problem (DCSP), P=⟨V, A, D, C⟩, consists of the following [17]:

- A set of n variables V ={x_1, x_2, …, x_n}
- A set of g agents A={a1, a2, …, ag}
- Discrete, finite domains for each of the variables D = {D1, D2, …, Dn}, Di = {d1,d2,. . .,d|Di| } for xi , i = 1,2,. . . ,n
- A set of constraints C={$C_1(x^1)$, $C_2(x^2)$, …, $C_m(x^m)$} ,that x^i , i=1,2,. . ., n, is a subset of x , and $C_i(x^i)$, determines the values that the variablesx^ican't take simultaneously. For example, a constraint as C ({x_1, x_2}) =⟨d_1, d_2⟩, the means is that: when x_1= $d_1$then $d_2$cannot assignment to $x_2$and when x_2= $d_2$then x_1 cannot take d_1.

The ultimate goal of solving the distributed constraint satisfaction is finding the values for assignment to variables that will satisfy all of constraints.

In a DCSP, each variable is owned by a different agent. The goal is still to find a global variable assignment that meets the constraints, but each agent decides on the value of his own variable with relative autonomy. While he does not have a global view, each agent can communicate with his neighbors in the constraint graph [18].

III. DESCRIPTION OF OUR PROPOSED METHOD

As noted earlier our proposed method considers the train scheduling as a DCSP then solving it in a multi agent system. The trains are variables that each agent owned one of these variables. The score of each agent is equal to traveled blocks by this agent. In first, the score of all agents are zero. All of agent start their work simultaneously (agents which is not their starting time from origin wait until be starting time). Each agent check the its required block in term of being free. If that block is free, lock this block and enter to it block. After traveling this block agent release this block and announce it to first agent in waiting queue for this block. Then this agent increase itself score and iterative this process to arrive to destination. But if require block was locked its agent request this block and enter to waiting queue. When the score of all agents is equaled to number of required block for arrival to destination, a solution is fined and algorithm terminated.

3.1. Our Proposed Method Algorithm

Input:number of trains, number of block, start time of each train, orientation of each train, Origin and destination of each train, the time required to traverse each block.

Output*:Arrival time to each block in the way to reach the destination for all trains.*

While (Score of all agents be equal to number of require blocks to reach the destination)

For all agents

If (start time of this agent <= current time) & (not arrived toDestination)

Start work of agent

for (blocki ϵ required blocks to reach the destination)

ifblocki is not locked

lockblocki and Travels its;

release time of blocki ← travel time blocki + current time

announce release time of blocki to other agents

agent score ← agent score + 1

else

requestblocki and be entered to waiting queue

wait for blocki until to release blocki

3.2. Example Of Algorithm Execution And Experimental Results

We implement this algorithm by MATLAB and for see the results, executed this algorithm for an example with the following characteristics.

- Example 1:

Number Of Block=12;
Number Of Train=7;
Name of Train=[T1, T2, T3, T4, T5, T6, T7];
Travel Time of Each Blocks =[35, 23, 75, 50, 45, 20, 15, 40, 20, 62, 19, 32];
Start Time Of Each Train=[0, 40, 80, 0, 40, 80, 120];
Start Station of Each Train =[0, 0, 0, 12, 12, 12, 12];
Direction of Each Train=[0, 0, 0, 1, 1, 1, 1]; 0 => North to South, 1 => South to North

- The obtained results of trains travelling without considering constraints are as fig. 2.

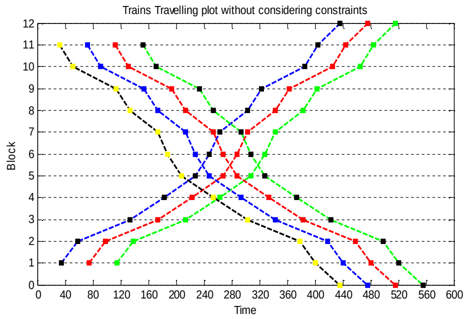

Fig.2. train travelling plot without considering constraints

As shown in Fig.2. Despite the small sample size of trains and blocks, there were number of crossing for trains at block 4 and block 7 in 3 incidents at least.

However, another run of the algorithm was performed to test the performance of the algorithm taking inconsideration the constraints.

Fig.3. shows the result of the implementation. As can be seen trains travel the required blocks without any cross.

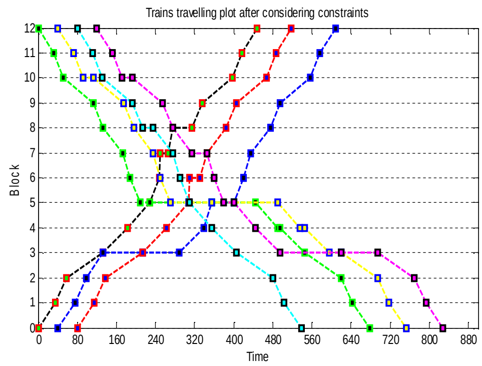

Fig.3. Trains travelling plot after considering constraint

The algorithm work to generate a timetable that organize the arrival time of each train at each block where no more than one train can be at a block at the same time. Table 1 shows the timetable generated as an output of the algorithm where the arriving time for each train is controlled, calculated and stated.

Table1. Arrive time of each train to each block

Block / Train	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12
T1	35	58	133	183	228	248	265	315	335	397	416	448
T2	75	98	288	338	400	420	435	475	495	557	576	608
T3	115	138	213	263	308	330	345	385	405	467	486	518
T4	678	643	620	545	490	208	188	173	133	113	51	32
T5	753	718	695	595	535	270	250	235	195	175	91	72
T6	112	131	193	213	275	290	310	355	405	480	503	538
T7	828	793	770	495	445	380	360	315	275	255	171	152

To verify the results obtained in the second test (described by in Fig.2.) and to confirm the efficiency of the algorithm in solving the problem of train scheduling, new test was designed to accommodate bigger number of trains and blocks, this test adopt a situation where 10 trains and 20 blocks applies. Fig.4. shows the result of the test implementation for 10 train and 20 blocks. As can be seen trains travel the required blocks without any cross in this case also.

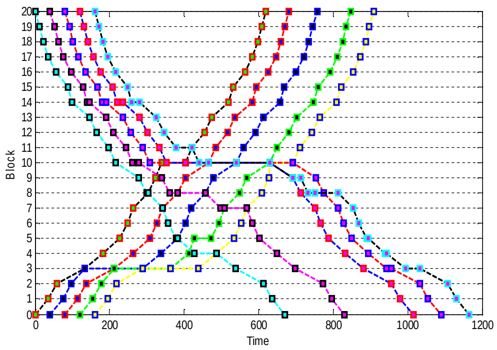

Fig.4. Trains travelling plot considering 10 trains and 20 blocks

3.3. Finding and future work

The distributed constraint satisfaction problem (DCSP) proved to be a reasonable and trusted technique to participate in the solution of train scheduling problem. The algorithm presented in this paper managed to create a time table whichcontains no flows (more than one train cross the same block at the same time) on two different size samples.

For future work, a real life situation or a much bigger sample of (trains, blocks, and trains and blacks) can be tested using the algorithm, then to analyze the obtained results in purpose of suggesting improvement for the algorithm.

CONCLUSIONS

This paper applies the distributed multi-agent system in responding to train scheduling problem. As multi agent systems managed to solve various complex problems, we found it is worthy examining the capabilities of it in solving the train scheduling problem. The contribution of multi agent system and DCSP managed to solve the issue and produced a flawless timetable to control the arrival of the trains to the dedicated blocks. The experiment were focused on two small to medium samples, a bigger sample or real life situation will be a higher level of validation to test the distributed multi-agent system capabilities in solving the problem, on the other hand test the performance of the multi-agent based solution against the current used techniques for train scheduling on a benchmarking basis. Hence, this shall put the distributed multi-agent system on the right track of optimization to produce multi-agent based solution for train scheduling problem in general.

REFERENCES

[1] Moreno, Hilda Beatriz Ramirez, et al. "Intelligent Agents as Support in the Process of Disease Prevention Through Health Records." KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications. Springer, Cham, 2018.

[2] Fioretto, Ferdinando, William Yeoh, and Enrico Pontelli. "A multiagent system approach to scheduling devices in smart homes."Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems.International Foundation for Autonomous Agents and Multiagent Systems, 2017.

[3] Grzonka, Daniel, et al. "Using a multi-agent system and artificial intelligence for monitoring and improving the cloud performance and security." Future Generation Computer Systems 86 (2018): 1106-1117, 2018.

[4] B. Szpigel, "Optimal Train Scheduling on a Single Line Railway," Oper.Res., vol. 72, pp. 344-351, 1973.

[5] R. L. Sauder and W. M. Westerman, "Computer Aided Train Dispatching: Decision Support Through Optimization," Interfaces, vol. 13, No. 6, pp. 24-37, 1983.

[6] R. G. Mills, S. E. Perkins and P. J. Paudney, "Dynamic Rescheduling of Long Haul Trains for Improved Timekeeping and Energy Conservation," Asia-Pacific J. of Oper. Res., vol. 8, pp. 146-165, 1991.

[7] D. R. Kraay and P. T. Harker, "Real-Time Scheduling of Freight Railroads," Transp. Res., vol. 29B, No.3, pp. 213-229, 1995.

[8] S. Tsutsui and D. E. Goldberg, "Search Space Boundary Extension Method in Real-Coded Genetic Algorithms," Info.Sci., vol. 133, pp. 229-247, 2001.

[9] Carey, Malachy, and Ivan Crawford. "Scheduling trains on a network of busy complex stations." Transportation Research Part B: Methodological 41.2, pp. 159-178, 2007.

[10] Yokoo, M., Durfee, E. H., Distributed constraint Satisfaction for formalizing distributed problem solving.12th IEEE International Conference on Distributed Computing Systems, 1, pp. 614–621, 1992.

[11] I. Jovanoski, D. Davcev, Distributed constraint satisfaction model adaptation for balanced energy, in: IEEE International Conference on Industrial Informatics,IEEE, 2006, pp. 636–640.

[12] K. Mizuno, Y. Fukui, S. Nishihara, Urban traffic signal control based ondistributed constraint satisfaction, in: Proceedings of the 41st Hawaii Inter-national Conference on System Sciences, IEEE, Hawaii, 2008.

[13] [13] R. Béjar, C. Domshlak, C. Fernández, C. Gomes, B. Krishnamachari, B. Selman,et al., Sensor networks and distributed

[14] H.W. Chun, Train timetable and route generation using a constraint-based approach, in: Tenth International Conference on Industrial & Engineering Applications of

Artificial Intelligence & Expert Systems, Goose Pond Press,Atlanta, Georgia, 1997, pp. 441–448.

[15] S.H. Semnani, K. Zamanifar, Using MaxCAPO to solve comprehensive university timetabling problem considering intercollegiate courses, in: 3th Joint Congress on Fuzzy and Intelligent Systems, Yasd, 2009.

[16] W. C. Zhong, J. Liu, M. Z. Xue and L. C. Jiao, "A multiagent genetic algorithm for global numerical optimization". IEEE Trans. on Systems, Man and Cybernetics (Part B), vol. 34, pp. 1128-1141, 2004.

[17] M. Yokoo and E. H. Durfee. "Distributed constraint optimization as a formal model of partially adversarial cooperation", Technical Report CSE-TR-101-91, University of Michigan, Ann Arbor, MI 48109, 1991.

[18] Y. Shoham, K. L. Brown, " MULTIAGENT SYSTEMS, Algorithmic, Game-Theoretic and Logical Foundations", 2009, 2010.

[19] Moghaddas, Raziyeh, EghbalMansoori, and Ali Hamzeh. "A new multi-agent algorithm for solving constraint satisfaction problems." In Information and Knowledge Technology (IKT), 2013 5th Conference on, pp. 197-201. IEEE, 2013.

[20] Tahmouresnezhad, Jafar, Mohammad BagherDastgheyb, FatemehAlizadehZoj, ElhamPejhan, RaziyehMoghaddas, and Ali Hamzeh. "An incomplete and asynchronous algorithm based on localization for Distributed Constraint Optimization." In Information and Knowledge Technology (IKT), 2013 5th Conference on, pp. 214-218. IEEE, 2013.

★★★