様　式　C－１９、F－１９、Z－１９　（共通）

科学研究費助成事業　　研究成果報告書

平成 ２８ 年　 ９ 月 ２１ 日現在

機関番号：６２６１５

研究種目：挑戦的萌芽研究

研究期間：2013～2015

課題番号：２５５４００９２

研究課題名（和文）リアルタイム難易度調整と分散制約最適化に基づく人の大規模運転行動データの収集

研究課題名（英文）Large-scale human behavior data collection based on real-time challenge balancing and distributed constraint optimization

研究代表者

ＰＲＥＮＤＩＮＧＥＲ　ＨＥＬＭＵ（Prendinger, Helmut）

国立情報学研究所・大学共同利用機関等の部局等・教授

研究者番号：４０３９０５９６

交付決定額（研究期間全体）：（直接経費）　　2,900,000円

研究成果の概要（和文）：我々はマルチユーザ３D環境下での環境配慮型運転訓練の為に開発したiCO2という持続的で環境配慮型交通管制ツールを実用化するために、マルチエージェントシステムを適用した。iCO2は信号機や周辺交通状況のコンピュータ制御によるエコドライブ訓練の支援を行う。それ故、各アージェントはそれぞれ対峙関係にあり、最適化された課題の達成に挑む。研究の目的は、この共有型シミュレーション空間で全ユーザに対する最適化された難易度調整を行うことであり、１）制御可能体（相手）と制御不可能体（ユーザ）があり、２）各々の相互関係が非常にダイナミックである。我々はこれを分配型制限最適化問題（DCOP）として問題解決した。

研究成果の概要（英文）：We apply multi-agent systems to realize sustainable eco-friendly traffic management We developed iCO2, an online tool for training eco-friendly driving in a multi-user 3D environment. iCO2 supports eco-driving practice by instructing computer-controlled agents, such as traffic lights and other vehicles, to create traffic situations that make eco-driving more difficult. Hence the agents take the role of “opponents” that try to achieve the optimal challenge level for the skill level of each user. The research challenge is to find the optimal challenge level for all user drivers in a shared simulation space that (1) involves both controllable entities (“opponents”) and non-controllable entities (users) and (2) is highly dynamic, with dependencies between entities being created and destroyed in real time. We solve this problem by modeling the scenario as a distributed constraint optimization problem (DCOP).

研究分野：AI

キーワード：環境配慮型運転トレーニング　最適化された難易度調整　分配型制限最適化問題（DCOP）

１．研究開始当初の背景

During the past decade, multi-agents systems (MAS) have been successfully applied to solve various types of sustainability issues. Examples include smart grid management, reduction of energy consumption in buildings, and traffic management. In traffic management, the goal is to minimize traffic congestion by creating optimal traffic control strategies. To solve traffic control problems, MAS techniques are used to synchronize adjacent traffic signals (conceptualized as agents), with the goal of letting vehicles cross as many intersections as possible without stopping. The difficulty is to determine which direction should be synchronized so as to maximize network throughput, and consequently minimize travel times and negative impacts of traffic on the environment.

While we share the goal of reducing fuel consumption and CO2 emissions with those MAS-based traffic control applications, the controllable agents in iCO2, our multi-user three-dimensional (3D) eco-driving training space, serve a completely different purpose. Rather than facilitating traffic flow, controllable agents act as “opponents” to the user driver by creating eco-circumstances (“obstacles”), i.e. traffic situations that make eco-driving difficult for the user. We implemented two types of opponents: fixed position and moving opponents. A traffic signal is an opponent (agent) with a fixed position that can impact all user drivers approaching it. As an opponent, the traffic signal can turn “yellow” at a certain moment and put the user driver in an “eco-safety dilemma” state: (1) strong braking or acceleration will avoid crossing the red light, but is an eco-unfriendly driving style; (2) keeping the speed is an eco-friendly option but has the risk of running the red light.

Computer-controlled vehicles, on the other hand, are moving opponents. Their behavior can impact the vehicles (both user-controlled and computer-controlled) following on the same lane. E.g. a braking operation will induce braking in the following vehicles, as collisions have to be avoided. The difficulty level of the eco-circumstance, or challenge, we want to create is determined by the user’s individual skill level. It is known that users get frustrated when the difficulty level of the interaction in the simulation space is too high and loose interest if it is too low. So the opponents (traffic signal, computer-controlled vehicles) should adapt the difficulty level to the user’s current skill level. This technique is called real-time, or dynamic, challenge balancing (RCB). Existing approaches for RCB are limited to the “one opponent vs. one user” case. Our training environment, on the other hand, is a “social” driving environment, where multiple users are expected to practice eco-driving in the same simulation space at the same time. Therefore, to create challenges for each individual user, multiple opponents are necessary.

To summarize, the technical challenge of our eco-driving environment is to find a set of actions for the opponents that generate the optimal challenge level for each user in the scenario.

A solution to this problem is complicated by several factors: (1) all entities (users and opponents) share the same driving space and thus their actions have direct and often unwanted influences on others. For instance, the attempt to create a high-difficulty eco-circumstance for one user driver may accidentally create a high level challenge for another user driver (following behind the first user), who should instead experience a low-difficulty eco- circumstance; (2) user drivers are non-controllable entities that might also create unwanted eco-circumstances for other users; (3) the solution has to be found in real-time. In this paper, we will represent our problem as a distributed constraint optimization problem (DCOP). In DCOP applications, each agent holds a variable and can change its value to achieve a globally optimal solution. To our knowledge, DCOP algorithms have not yet been applied to scenarios that are comparable to the heterogeneity of our multi-user multi-opponent real-time eco-driving scenario. Hence, the main contribution of this project is a new application for DCOP algorithms and the presentation of results from a series of simulation runs that involve human users, a kind of “participatory simulation”.

２．研究の目的

Multi-agent systems have already been successfully applied to a variety of traffic control problems and demonstrated the potential to lower travel times and environmental impact. Sharing this goal, we have developed iCO2, an online tool for training eco-friendly driving in a multi-user three-dimensional environment. iCO2 supports eco-driving practice by instructing computer-controlled agents, such as traffic lights and other vehicles, to create traffic situations that make eco-driving more difficult. Hence the agents take the role of “opponents” that try to achieve the optimal challenge level for the skill level of each user. The research challenge is to find the optimal challenge level for all user drivers in a shared simulation space that (1) involves both controllable entities (“opponents”) and non-controllable entities (users) and (2) is highly dynamic, with dependencies between entities being created and destroyed in real time. We try to solve this problem by modeling the scenario as a distributed constraint optimization problem (DCOP). The main contribution of our paper is the application of a DCOP algorithm to such a new type of application scenario. We evaluate our approach by running scenarios both in terms of speed and optimality of the solutions proposed by the DCOP algorithm.

３．研究の方法

In distributed constraint optimization problems (DCOP), each agent is assigned to one or more variables and these have interdependencies. The goal is to find an optimal assignment for the variables to minimize or maximize a global cost function.

A DCOP is formally defined by (following the notation of:

A DCOP is represented by a constraint graph, where the nodes are the variables and a link between two nodes exists when there is a constraint over two variables. Agents whose variables share a constraint are called “neighbors”. Agents are only allowed to see their neighbors, and may exchange messages with them.

In our application, each agent represents either an opponent or a user. In the case of an opponent, the variable associated with it indicates the action that the opponent will choose to execute. Since the users are non-controllable entities, the variable indicates the action that we predict the user to execute.

The prediction is based on (1) the current environment state (car's position, acceleration, speed, and distance to the light) and (2) the actions that the car ahead may perform (i.e. the value of the parent node). It is important to note that not all opponents involved in the DCOP may influence the prediction of a certain user's action. To reflect this, we introduce the concept of "dominance". An agent (opponent or user) is dominant over another agent (opponent or user) if the actions of the former directly influence the behavior of the latter. In our application:

1. A traffic signal is dominant over an agent Ai, if Ai is in front of the traffic light and there is no other agent Aj in the environment between Ai and the traffic light.
2. An opponent/user car is dominant over an agent Ai, if Ai is behind the opponent/user car, in the same lane, and there is no other agent Aj in the environment between Ai and the opponent/user car.

An opponent is called *non-controlling* if it is not dominant over any other agent (at a particular moment). Non-controlling opponents are not part of the constraint graph, since they cannot influence any user. In our scenario, DCOPs always involve a traffic light. Hence, we obtain a tree-like structure, where the traffic light is the root, each branch represents a lane, and the leaf nodes are users (see Figure 2 (right)). This means that if agent Ai is dominant over agent Aj, the node controlled by Ai is a parent node of the node controlled by Aj. Hereafter, we refer to the nodes using tree notation (parent node and child node). Once the structure is determined from the position of the traffic lights, opponent cars and users in the 3D environment, we can start with distributed constraint optimization.

The domain of a variable in the DCOP depends on the type of the agent controlling that variable. If the variable corresponds to a traffic signal node, there are two available actions, "change color" and "not change color": D = {C, NC}. If the variable corresponds to an opponent car node, the available actions are braking, keeping the speed and accelerating: D = {B, KS, A}. In the case of the user node, the domain is the output of the prediction function, which is also "braking", "keeping speed", or "accelerating": D = {B, KS, A}. Even though the user is allowed to change lanes, this is not covered by the prediction function.

The existence of constraints is solely based on the dominance relationship between agents. The cost of existing constraints depends on whether there is a conflict between the agent's actions or not. A conflict occurs when (a) a parent node (dominant agent) executes an action that raises an inadequate challenge for the child node (in case it is a user node); (b) a child node executes an action that is conflicting with its parent node's value. As an example of the latter type of conflict, consider an opponent car that is in front of a traffic light and decides to accelerate, while the light decides to change to red. If the car is very close to the light, it may be able to accelerate and cross in time; hence both actions can be executed independently. However, if the car is slightly further away, it may be forced to brake, even if it had chosen to accelerate (recall that opponents never cross a red light). This unwanted braking will affect the following cars and hence, this kind of conflict should be detected beforehand.

Case 1: When the child node xj is an opponent car, we predict the action of the opponent car given the value of the parent xi, and the current state. Then we compare the predicted action with the desired action (dj, the value of xj). This comparison is based on the restrictiveness of the action, "brake" being the most restrictive action and "accelerate" being the least restrictive one. If the predicted action is more restrictive than the desired one, they are conflicting, and the constraint has infinite cost. Otherwise, there is no cost (f (xi, xj) = 0).

Case 2: When the child node xj is a user, we similarly predict the action that the user car will execute, given the value of the parent xi and the current state. Furthermore, we predict the challenge that this action represents by using the challenge function. If the predicted action is different from the value of the child node xj, there is infinite cost. This is necessary because the DCOP algorithm investigates all the possible values for each node. Since the user node should always represent a predicted action, values that are inconsistent with it should yield infinite cost. If the predicted action is the same as the value of the child node xj, the cost is the difference between the predicted challenge and the ideal challenge for that user. The ideal challenge for a user is the difference between the maximum challenge and the current average challenge for the user.

４．研究成果

The results of our pilot study are summarized in Table 1. The traffic signal is included in the size of the DCOP (e.g. a problem of size 7 should have one signal and 6 cars). As expected, when the number of entities in the scenario grows, the number of DCOPs to be solved also grows, as well as the size of the DCOP. Even though 33 computer-controlled cars and 3 human users shared the same circuit in Session III, the DCOP maximum size did not exceed 11 variables (traffic light included).

The study showed that regardless of the growth in size, the average time taken to solve the DCOPs remained approximately constant (1 second). Even if 1 second appears slow for graphs of such small size, it is important to note that this value does not represent solely the algorithm runtime. Since all of the graphics and logic complexity of the application runs in a single thread, the exchange of messages between agents has a natural delay, even if it runs locally. Hence, the actual time consumed by the DCOP algorithm might be much smaller than 1 second.

As a metric for the level of "dynamics" (rate of change of the position of user cars) of the traffic situation, we counted the number of reconstructions of the DCOP graph due to lane changing. First, the graph is always reconstructed when a car crosses the traffic signal (i.e. leaves the graph). It would also be reconstructed if a vehicle enters the relevant area. However, such situation was not foreseen in our experiment. Second, the lane changing action induces a reorganization of the existing variables and indicates how "dynamic" did the user cars behave, when inside the DCOP area. However, while users were allowed to freely move within the scenario, few graph reconstructions were detected in each session (see Table 1).

Although challenge balancing was not the focus of this study, we also looked at whether the DCOP calculations have any effect on the value of the challenge function over time. Fig. 3 shows the example for one user in Session II. For this experiment, the challenge function was weighted so that it could assume values within the range [0-4]. The peak at 2min50s indicates a vehicle collision (challenge value raises because of the collision itself, and the sudden deceleration generated by it). The figure shows that, overall, the challenge for this user remained low throughout the whole session. This could indicate many things: opponents' behavior is not aggressive enough, user was not affected by challenge balancing (as this would have yielded a generally higher challenge, in average) or that the challenge function needs to be adjusted. Further investigation is necessary to establish a direct relation between the participation in the DCOPs and the change in the value of the challenge function.

５．主要な発表論文（研究代表者、研究分担者及び連携研究者には下線）

Helmut Prendinger, Kamthorn Puntumapon, and Marconi Madruga, Extending real-time challenge balancing to multiplayer games: A study on eco-driving. *IEEE Transactions on Computational Intelligence and AI in Games*, Vol. 8, No. 1, 2016, pp 27-32

Marconi Madruga and Helmut Prendinger., iCO2: Multi-user eco-driving training environment based on distributed constraint optimization. *Proc 12th Int'l Conf on Autonomous Agents and Multiagent Systems (AAMAS'13)*, Saint Paul, Minnesota, 2013.5, pp 925-932.

Marconi Madruga and Helmut Prendinger, iCO2 - Promoting eco-driving practice through multiuser challenge optimization (Demo). *Proc 12th Int'l Conf on Autonomous Agents and Multiagent Systems (AAMAS'13),* Saint Paul, Minnesota, USA, 2013.5, pp 1381-1382.

〔雑誌論文〕（計　3 件）

〔学会発表〕（計　0 件）

〔図書〕（計　0 件）

〔産業財産権〕
○出願状況（計　0 件）

名称：
発明者：
権利者：
種類：
番号：
出願年月日：
国内外の別：

○取得状況（計　0 件）

名称：
発明者：
権利者：
種類：
番号：
取得年月日：
国内外の別：

〔その他〕
ホームページ等

http://research.nii.ac.jp/~prendinger/#

６．研究組織

(1)研究代表者
プレンディンガー　ヘルムト
（PRENDINGER, Helmut）
国立情報学研究所・コンテンツ科学研究系・教授
研究者番号：40390596

(2)研究分担者
（　　　）

研究者番号：

(3)連携研究者
（　　　）

研究者番号：