Design and Implementation Scheme of an Individual Game Support System Driven by High-Frequency Data

Andrzej M.J. Skulimowski
AGH University of Kraków, Decision Sciences Laboratory
and Progress & Business Foundation
Kraków, Poland *ams@agh.edu.pl*

Abstract

This article contains a summary description of a decision support system aimed at analysing ball tracking data and deriving real-time recommendations to players of individual games such as minigolf. We review the best practices of game support methods and propose the software architecture of a game support system (GSS). The system is based on real-time analysis of high-frequency ball movement data. It uses a dynamic programming-type algorithm to compute optimal moves from the current position of the player, and provides hints by accounting for the hitherto score and game level of the player.

Keywords: Decision support systems, High-frequency data, Minigolf, Game management

1. Introduction

Individual game support systems are an emerging area in the field of information systems development. Players in field games such as tennis, golf or minigolf (miniature golf) are increasingly supported by AI techniques that cover a diversified spectrum of gaming activities: from real-time recommendations communicated via a mobile app, off-line game analysis and training planning, to supporting courses and game premises managers [7]. Furthermore, hints to the course designers can be derived from game statistics. The main aim of the game support system (GSS) presented in this article is to gather and process information received from a visual ball tracking system (BTS) in order to outline optimal strategies adjusted to individual features and needs of players. The BTSs contain sensors distributed over the playground as well as inside the balls and putters. The GSS hints can either be communicated to players in real-time or be used offline to analyse the games and identify potential improvements. The ultimate goal is to ensure competitiveness of game station designers and manufacturers by alignment to recent trends and to players' preferences as regards mobile app game support.

The architecture and implementation of the GSS presented in this brief article presumes a prior catchment of visual ball tracking data which will be analysed as GSS inputs. Therefore, the GSS will cooperate with an external visual tracking system and has to adjust its analysis mode to the quality, frequency, and type of data. Additional inputs to the game analytics can be provided by the simultaneous gathering of other physical data, such as force applied and putt approaches used by players to hit the ball. Needs analysis showed that tracking ball trajectories with high frequency and analysing the data captured in real-time is sufficient to feed the GSS. The tracking of ball bouncing seems to be a particularly difficult problem due to the changing ball direction after a bounce.

Characteristic features of minigolf that make it different from classical golf are the way of movements, the possibility of rolling the ball after a bounce, and the array of objects that may be inserted in the course by its designers. The ball may disappear in tunnels or pass through different obstacles etc. All these features and circumstances must be accounted for in the ball movement models used in the GSS.

This brief paper is organised as follows. Section 2 contains the basic notions used to formulate problems solved by players during the game. Section 3 is devoted to the statement of a strategy optimization problem solved by a GSS. The GSS architecture and implementation plan are outlined in Section 4. In Section 5 we present our conclusions.

2. Basic definitions and related work

The principal notion of the current research is a game, which is described as a sequence of actions of one or more players from a certain community $\mathbb{P}$ and their action assessments. A *player* is characterised by a unique label $P \in \mathbb{P}$, a vector of *skills* $u(P)$ relevant to the game, and the personal characteristics provided voluntarily to the system, such as age, sex, height, or weight. The *object* of the game is an entity characterised by an individual identifier Q, the state s from a finite set of states S, the physical position in a certain coordinate system, say $x:=(x_1,x_2,x_3)$, and the values of other features such as weight, colour. If the item Q is a standard minigolf ball, then in a game involving one player, it is fully characterised by the state, velocity, and coordinates of its barycentre. An object is in the *stable state* if it does not move. The set of admissible states may also include special states such as „*broken*”, „*lost*”, or *loops L*. If an object is in any of the latter (“*loop*”) states, its movement is perpetual and its position cannot be described by one coordinate vector. Only a few minigolf game stations (and none of the golf holes) allow an object to reach a loop state. An *action* is characterised by the attribute vector $a_t=(a_{t1},\dots,a_{tn})$ applied by a player P to an object Q in the state s_t and position x at time t. The state of an object may change as a result of a player’s action according to the formula

$$s_{t+1}(Q)=\delta(s_t(Q),a_t,t), \tag{1}$$

where δ is a game-specific state transition function. An application of an action will be termed a *move*, which is usually related to one *putt* (ball stroke). Observe that the eq.(1) above is a description of a controlled discrete event system (DES, [10]).

The minigolf game with a single course starts from an initial position that is common for all players. After each move of a player P who activated the GSS app installed on a personal mobile device, the GSS engine calculates the assessment of this move f_1, and generates the assessment f_2 of the current game situation. The latter assessment may be represented as a probability distribution of a random variable describing the number of moves needed to reach the end hole by the game object. The game terminates when the object reaches its target, i.e. the assessment f_2 of the current game object is zero. The game may also terminate when a prescribed number of moves was made, if the time allotted to this game was exhausted, or if the player abruptly withdraws. More termination conditions may be defined when needed, for example the termination of player P_1’s game may be contingent on reaching a prescribed game result by another player P_2. A schematic representation of minigolf game with visual ball tracking is shown in Fig. **1**.

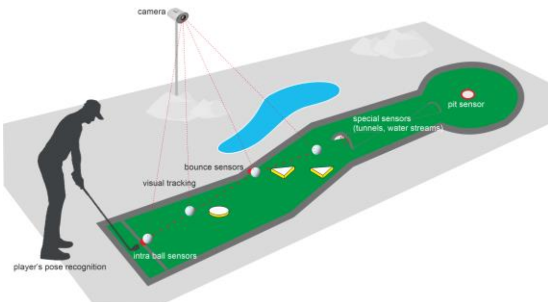

Fig. 1. A scheme of data capture during a minigolf game.

Below we define key notions that are used in the analysis of players’ moves in a GSS.

Definition 1. *An event in a multistep game is any change of the playing object state.* ■

Definition 2. *The trajectory of an object Q is a sequence of interlaced events and moves that start and end at a stable state. A move refers to the change of the physical position of an object Q.* ■

To every *player* $P \in \mathbb{P}$ there may be associated a *history* $H(P)$ of games played so far. In this context, history refers to the database that consists of previous game records, including time and duration of play, the course and holes where the game took place, the timeline of all events and moves with the ball trajectories and their assessments. The history can be updated either automatically in real-time while the game is being played and the GSS app is turned on, or manually after the game, if the player recorded or memorised the course of previous games. History is crucial to activate machine learning (ML) algorithms that generate real-time hints or off-line training programmes. Henceforth, whenever we refer to golf or minigolf, the objects will be termed *balls*.

Finding optimal strategies in minigolf has been studied in the literature within the framework of forward sequential planning. In [2], [3], [3] the authors analysed the game from the point of view of an optimal player, without taking into account various game skill levels when generating strategies. General approaches to golf or minigolf strategy planning can be found in [6], [8]. These approaches need to be adapted to a specific minigolf course, hole configurations, game variants, maximum allowable number of simultaneous players, the targeted DSS implementation, and other circumstances. Another area of related research is game robotics, where robots learn minigolf [4].

3. Optimal putt strategy problem and its solution

An optimal strategy problem in one-move minigolf game can be broken down in three subsequent tasks. The first task can be formulated as the following:

Problem 1. For a given player P with skills $u(P)$ and for the ball in state and position (s,x) calculate the parameters $(a_1,\dots,a_n)$ of an action a which is optimal from the point of view of the assessment f_2 of the state s_{n+1}. ■

Given the transfer function δ in eq. (1), Problem 1 becomes a nonlinear mixed continuous-discrete optimization task that can be solved with an appropriate combinatorial programming method applied to state transition merged with a continuous gradient minimization. The latter will be applied to the parameters of ball movement resulting from an action and to the final position of the ball after this move with an assumed value of u.

Before passing to the solution of the above-stated Problem 1, we will define the visual BTS V as a set of devices performing a joint measurement procedure of golf ball putt parameters by the players and a series of snapshots of a ball's trajectory following each putt till the ball reaches its stable state s_{n+1}. In real-life games, the ball may contain internal sensors V_1, thus being able to transfer the force measurement results to a receiver. The second component V_2 of V is a system of cameras or other visual sensors that follow the ball movement and retrieve its temporary positions during the move. They are also capable of identifying the stable final position of a ball which corresponds to reaching the state s_{n+1}. The third component of V is the system of sensors V_3 distributed over the game area and acting according to Internet of Things (IoT) principles. These sensors confirm the assessment $f(Q,a)=1$, which refers to a situation when the ball is in the final hole. The sensors can also retrieve the coordinates of a stopped ball to confirm the results of the last tracking or find a lost ball when no recent tracking information is available. V_3 can measure players' movements in order to recommend reaching an optimal position and pose before the next putt.

In real-life situations, the parameters of δ are unknown and must be estimated from observed ball trajectories that are considered as results of measured action parameters. This is accomplished by solving the following visual tracking system design problem.

Problem 2. Minimize the cost of V and the real-time estimation error of the ball trajectory averaged over a set of potential hits and initial states of the ball. Then, learn the player P skills $u(P)$ and derive the parameters of the transition function δ from the available ball trajectory measurements and skills estimation with minimum error. ■

When an optimal single putt strategy is calculated within Problem 1 and the player P skills $u(P)$ are learned as part of Problem 2, another problem needs to be solved, namely:

Problem 3. For player P with skills $u(P)$ find the minimum number n of putts to reach the final destination B from the position x and ball state s with probability p or more. ■

The putt planning algorithm is based on dynamic programming (DP); however, the classical DP is not applicable as exemplified in Fig. 2. A suitable procedure is proposed as Alg. 1 below. The reachability zone $Z(y,p,P)$ of y is the subarea of the game station such that player P with skills $u(P)$ can reach y in one putt with a probability at least p.

Algorithm 1 (solution to Problem 3)

```
input: s, x, u(P), p, nmax, n←1, Z←empty
while x not in Z do
   calculate the edge points C(Z) of Z  //C(Z) are extreme points of the convex
   if x is in Z return n, p;           //*hull of current approximation of Z
   for every edge point c in C(Z)
      calculate the reachability zone Z(c) of c
      set Z←Z ∪ Z(c);
      if x is in Z return n+1, p
   end (for)
   n←n+1
end (while)
if nmax≤n return ‘solution not found’   // applying computing time limit
end
```
■

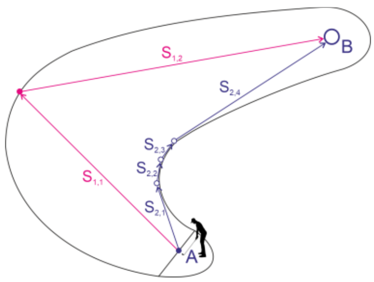

Fig. 2. An example of an optimal putt strategy in minigolf (2 putts to reach the hole B from A: $s_{1,1}$,$s_{1,2}$) compared to a non-optimal shortest path strategy (4 putts to reach the hole B from A: $s_{2,1}$,$s_{2,2}$,$s_{2,3}$,$s_{2,4}$).

4. Game decision support design

The essence of real-time player support is to achieve the game goal in an optimal way, while preserving individual player constraints resulting from physical abilities or handicaps. If several players strive to achieve a common goal with or without the rivalry from another team, the tasks of the GSS will extend to group support and coordination, while individual support will be offered without disturbing the optimality of the joint task.

The players provide all or only a part of the above information. Then, the GSS will generate the corresponding recommendation concerning the current move (putt parameters) or a series of moves, depending on the user-defined recommendation options. Observe that the class of limits of optimality has merely a technical character and ensures that the GSS recommendations are realistic. The procedure may be automatically or manually activated by the user after each putt. The decision analytics will apply reinforcement learning (RL) to learn the parameters of the recommendation generator. The RL outcomes may be enhanced with intertemporal supervised learning during the single game periods when the game instructor inspects and corrects the recommendation parameters.

The software architecture of the present version of game support and game management systems covers player support problems. We propose its further development prospects towards a managerial and manufacturer support system to yield an extended class of Game Management Systems (GMS). The analytic engine of the GSS should contain at least the following modules:

- Predictive analytics [9], including Bayesian networks and autoregressive models
- Multicriteria decision making [11] such as reference set and causal models
- Player recommendation generation and game training modules [1] applying the user skills and game history records stored in the GSS knowledge base [3]
- User community building with social network and controlled cellular automata.

A characteristic feature of the GSS and GMS is their modular architecture that allows the course managers to gradually develop it. The data is pre-processed in the GSS analytic engine with simple arithmetic averaging and the aggregation transform (cf. [12]).

The knowledge base that stores game history records merged with *ex-post* game assessments is a core GSS module. The history records are the fundamental input data to semi-supervised ML procedures, where only partial labelling is possible, while games performed on brand new courses with novel holes might still be analysed by the system. Multicriteria optimization and decision processes apply the above defined game criteria.

5. Conclusions

In this brief paper we presented Game Support Systems (GSS) as a new class of real-time DSS with challenging application prospects. Among the new research questions we formulated and solved a novel variant of adaptive dynamic programming that minimizes the number of moves to reach the goal of an individual game rather than any physical indicator. A pivotal role in the GSS is played by learning user skills retrieval with ML methods.

The architecture of the above proposed example of a GSS takes into account the best practices of information system development as well as a state-of-the-art approach in the physical planning of minigolf courses. We proposed a novel recommendation methodology which will enable multiple player support during individual ball games via a mobile app in real-time. It is based on two pillars: first, the efficient filtering out irrelevant ball motion data with monotonic aggregation, which speeds-up the data analysis process. Second, a backward planning approach that allows the GSS to find optimal putt sequences for any current location of the ball. An implementation work plan offers a realistic prospect for the GSS deployment within a competitive business model.

References

1. Couceiro, M.S., Portugal, D., Gonçalves, N., Rocha, R., Luz J.M.A., Figueiredo, C.M., Dias, G.: A methodology for detection and estimation in the analysis of golf putting. Pattern Anal Applic 16, 459–474 (2013). https://doi.org/10.1007/s10044-012-0276-8
2. Ge X., Lee J.H., Renz J., Zhang P.: Hole in one: Using qualitative reasoning for solving hard physical puzzle problems. In: Proc. 29th International Workshop on Qualitative Reasoning, New York, p.8. (2016). https://doi.org/10.3233/978-1-61499-672-9-1762
3. Ishikawa, M.: High-Speed Vision and its Applications Toward High-Speed Intelligent Systems. Journal of Robotics and Mechatronics 34(5), 912-935 (2022).
4. Khansari-Zadeh, S.M., Kronander, K., Billard A.: Learning to play minigolf: A dynamical system-based approach. Advanced Robotics 26(17), 1967–1993 (2012).
5. Khurana, H., Bombile, M., Billard, A.: Learning to Hit: A statistical Dynamical System based approach. In: IEEE Int. Conf. on Intell. Robots and Systems, pp. 9415-9421 (2021).
6. Kronander K., Khansari-Zadeh M.S.M., Billard A.: Learning to control planar hitting motions in a minigolf-like task. In: IEEE Int. Conf. IROS, pp. 710-717 (2011).
7. Lei, X., Humphreys, B.R.: Game importance as a dimension of uncertainty of outcome. J. Quantit. Analysis in Sports 9(1), 25-36 (2013). https://doi.org/10.1515/jqas-2012-0019
8. Markwell L.T., Cochran K., Porter J.M.: Off the shelf: Investigating transfer of learning using commercially available virtual reality equipment. PLoS ONE 18(10), e0279856 (2023). https://doi.org/10.1371/journal.pone.0279856
9. McCarthy, R.V., McCarthy, M.M., Ceccucci, W.: Applying Predictive Analytics. Finding Value in Data. Springer Nature Switzerland AG, Cham, p. 274 + XV (2022).
10. Skulimowski A.M.J.: Optimal Control of Asynchronous Discrete-Event Systems. In: Automatic Control in the Service of Mankind. Proc. of the 11th IFAC World Congress, Tallinn (Estonia), Aug. 1990, Vol.3. IFAC Proc. Volumes 23(8 Part 3), 489-495 (1991).
11. Skulimowski, A.M.J.: Freedom of Choice and Creativity in Multicriteria Decision Making. In: 5th Int. Conf. on KICSS, Chiang Mai, Thailand. Lect. Notes in Artif. Intell. 6746, pp. 190–203, Springer (2011). https://doi.org/10.1007/978-3-642-24788-0_18
12. Skulimowski A.M.J.: Revealing complexity-related time-series features with the Monotonic Aggregation Transform. In: 2014 IEEE 26th Int. Conf. on Tools with Artificial Intelligence, Limassol, Cyprus, pp. 694–700 (2014). https://doi.org/10.1109/ICTAI.2014.109