An Optimization Model for the Game "Toward the Stars"

October 25, 2014

1 Introduction

Toward the Stars (TtS) is a cooperative First-Person Action Game with emphasis on spaceship management gameplay. Four players engage in 10-15 minute long missions where they must deal with various events that take place during their starship adventures. The objective of the game is to get to destination and survive all the hazards met during the journey.

This research concerns the definition and optimization of mathematical models that represents the basic mechanics of TtS. Their solutions provide the sequence of operations that the players should take to solve the game in the best possible way. Knowing the best (or a good) solution allows for the extraction of a number of game measures, such as interestingness [1], uncertainty [2], and interaction [3]. These attributes find application in the design phase of the development of the game. In fact, they provide a feedback to game designers regarding many aspects of the game. In the academic literature, automated measurements of games have been used to devise Evolutionary Game Design methods [4], i.e., algorithms capable of designing interesting and challenging games. Similarly, the optimization model for TtS could be the first step toward the definition of a system that automatically defines the game, according to a specified level of challenge. Additionally, the solution identified by the model can be used as a basis of comparison for the players' actions while the game is running. This, in turn, allows for further analyses, such as assessing their level of skill and cooperation.

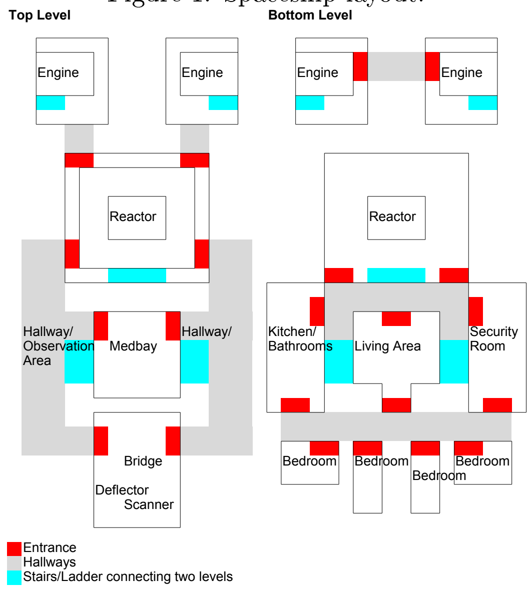
2 Structure of the Game

TtS is a complex game that is comprised of numerous features. In this research we focus on the most relevant of these features, so as to obtain a simplified representation of the game that can be modelled mathematically. The game takes place in the players' spaceship. The layout of the spaceship is illustrated in Figure 1. Different types of hazards are possible in the game, such as asteroids, fire, broken equipment, and intruders. Also, the players can use the following tools: laser gun, fire extinguisher, repair tool, and healing spray. Finally, by accessing a special interface the players can use the turrets to shoot down incoming asteroids.

In summary, the game can be segmented into the following basic tasks, each composed of several operations:

- **Getting rid of an intruder**. This task requires the following operations: deciding how many players take part in the task, assigning the availabe weapons to the participating players, getting the par-

Figure 1: Spaceship layout.



ticipating players to the assigned weapons, getting the armed player to the intruder location, and fighting.

- **Putting out a fire**. This task requires the following operations: deciding how many players take part in the task, assigning the availabe fire extinguishers to the participating players, getting the participating players to the assigned fire extinguishers, getting the equipped player to the fire location, and putting out the fire.

- **Fixing an equipment**. This task requires the following operations: deciding how many players take part in the task, assigning the availabe repair tool to the participating players, getting the participating players to the assigned repair tools, getting the equipped player to the equipment location, and fixing the equipment.

- **Healing a player**. This task requires the following operations: deciding how many players take part in the task, assigning the availabe healing spray to the participating players, getting the participating players to the assigned healing spray, getting the equipped player to the hurt player location, and healing the hurt player.

- **Defending the ship from incoming asteroids**. This task requires the following operations: deciding how many players take part in the task, assigning the availabe turrets to the participating players, getting the participating players to the turrets, and shooting down the asteroids.

It can be easily seen that all the tasks have a very similar structure. In general, the set of operations that comprise all the task is:

- Deciding how many player take part in the task.

- Assigning limited resources to the players.

- Identifying the best way to go from a location to another.

- Specific operations to solve the task.

Given the similarity among the tasks, we decided to focus on solving a task and then extending the methodology to the others. Specifically, we addressed and solved the problem of putting out a fire, as shown in the following section.

3 A Mixed Integer Optimization Model

In this section we propose a Mixed Integer Problem (MIP) for the task of putting out a fire. A MIP is a mathematical model comprised exclusively of:

- Linear equations and inequalities.
- Integer, binary, and continuous variables.

A MIP model is easier to solve than non-linear models and, generally, can be solved directly by means of general optimization packages such as GAMS [5], instead of relying on ad hoc algorithms. Also, modelling a problem provides several insights on its structure that can be exploited to devise faster and more effective methodologies.

In the following, the elements comprising the model are presented and explained.

3.1 Sets

$G = (V, A)$ Graph representing the layout of the ship.

V Set of vertices. Indexed by i, j.

A Set of arcs. Indexed by (i, j).

P Set of players. Indexed by p.

E Set of extinguishers. Indexed by e.

K Number of players. Indexed by k.

3.2 Data

v_p Vertex location of player p.

v_e Vertex location of extinguisher e.

v_f Vertex location of fire f.

vcomp (i, j) Function that returns 1 if i is the same vertex as j.

t_{ij} Time taken to go from i to j.

d_i Damage taken by a player visiting i. It is assumed that players getting into a room on fire without a fire extinguisher will take damage. On the other hand, players equipped with a fire extinguisher will not get any damage.

τ_k Time required by k player to completely extinguish the fire.

D Maximum damage sustainable by a player.

3.3 Variables

n_k Number of players variable. Equals 1 if k players participate to the task.

c_p Task participation variable. Equals 1 if player p participates to the task.

m_{pe} Player to extinguisher assignment map. Equals 1 if extinguisher e is assigned to player p.

x_{ij}^p Player p path variable to extinguisher. Equals 1 if player p traverses arc (i, j) while going to retrieve the extinguisher.

y_{ij}^p Player p path variable to fire hazard. Equals 1 if player p traverses arc (i, j) while going to retrieve the extinguisher.

T^p Total time spent by player p.

W^T Worst (greatest) total time spent.

D^p Total damaged suffered by player p.

3.4 Constraints

Number of Players Constraints

$$\textstyle\sum_{k=1}^{K} n_k = 1 \tag{1}$$

$$\textstyle\sum_{p \in P} c_p = \sum_{k=1}^{K} k n_k \tag{2}$$

$$n_k \in \{0, 1\} \qquad \forall k = 1, \ldots, K \tag{3}$$

Constaint (1) forces to select exclusively one number of players. Constraint (2) defines what number of players is participating in the operations.

Task Participation Constraints

$$\textstyle\sum_{p \in P} c_p \geq 1 \tag{4}$$

$$c_p = \textstyle\sum_{e \in E} m_{pe} \qquad \forall p \in P \tag{5}$$

$$|A| \, c_p \geq \textstyle\sum_{(i,j) \in A} x_{ij}^p \quad \forall p \in P \tag{6}$$

$$|A| \, c_p \geq \textstyle\sum_{(i,j) \in A} y_{ij}^p \quad \forall p \in P \tag{7}$$

Constraint (4) imposes that at least 1 player must participate in the task. Constraints (5) states that a player that participates must be assigned to a fire extinguisher and viceversa. Constraints (6) and (7) link the player participation variables to the path variables.

Player/Extinguisher Assignment Constraints

$$\textstyle\sum_{e \in E} m_{pe} \leq 1 \quad \forall p \in P \tag{8}$$

$$\textstyle\sum_{p \in P} m_{pe} \leq 1 \quad \forall e \in E \tag{9}$$

$$m_{pe} \in \{0, 1\} \quad \forall p \in P, \forall e \in E \tag{10}$$

Constraints (8) and (9) impose that a player can be assigned at most to a fire extinguisher, and viceversa.

Player to Extinguisher Path Constraints

$$\sum_{j \in V} x_{ji}^{p} + \text{vcomp}\,(i, v_p)\, c_p = \sum_{j \in V} x_{ij}^{p} + \sum_{e \in E} \text{vcomp}\,(i, v_e)\, m_{pe} \quad \forall p \in P, \forall i \in V \tag{11}$$

$$0 \leq x_{ij}^{p} \leq 1 \quad \forall p \in P, \forall\,(i,j) \in A \tag{12}$$

Constraints (11) define the path of a player from his/her initial location to the assigned fire extinguisher location.

Extinguisher to Fire Path Constraints

$$\sum_{j \in A} y_{ji}^{p} + \sum_{e \in E} \text{vcomp}\,(i, v_e)\, m_{pe} = \sum_{j \in A} y_{ij}^{p} + \text{vcomp}\,(i, v_f)\, c_p \quad \forall p \in P, \forall i \in V \tag{13}$$

$$0 \leq y_{ij}^{p} \leq 1 \quad \forall p \in P, \forall\,(i,j) \in A \tag{14}$$

Constraints (12) define the path of a player from the location of the assigned fire extinguisher to the location of the fire.

Attributes

$$T^{p} = \sum_{(i,j) \in E} t_{ij} x_{ij}^{p} + \sum_{(i,j) \in E} t_{ij} y_{ij}^{p} + \sum_{k=1}^{K} \tau_n n_k \quad \forall p \in P \tag{15}$$

$$W^{T} \geq T^{p} \quad \forall p \in P \tag{16}$$

$$D^{p} = \sum_{(i,j) \in E} d_{ij} x_{ij}^{p} \quad \forall p \in P \tag{17}$$

$$D^{p} \leq D \quad \forall p \in P \tag{18}$$

Constraints (15)-(17) calculate the time taken by each player to extinguish the fire, the worst (largest) time, and the damage taken by each player. Constraint (18) imposes that the damage received by a player cannot exceed D.

Objective Functions

$$\min \quad W^{T} \tag{19}$$

The objective of the model is to minimize the maximum time taken by a player to extinguish the fire.

Discussion

The model is comprised by a large number of binary and continuous variables. This is due to the fact that the model as a whole is comprised by a set of subproblem, as explained in Section 2, and therefore many variables and constraints are required to link and connect its different parts.

In the literature on optimization it is a well-known fact that, when possible, it is more efficient to solve many smaller sub-problems rather than a single bigger problem, as the complexity of these problems grows exponentially. In the next section we present an optimization algorithm that exploits the structure of the problem to solve the problem in a very efficient way.

4 A Decomposition Algorithm

As explained in Section 2, the problem of putting out a fire is comprised of the following steps:

1. Deciding how many players take part in the task.
2. Assigning the available fire extinguisher to the participating players.
3. Moving the participating players to the locations of the assigned fire extinguishers.
4. Moving the players from the locations of the assigned fire extinguishers to the fire location.
5. Putting out the fire.

Obviously, every step is interconnected to the others. However, we can solve each subproblem indipendently and then merge the solutions to find the single best solution to the global problem. In the following, the element comprising the decomposition algorithm are explained.

4.1 Putting Off the Fire

In terms of the model, putting out the fire is not real operation as it only affects the total time taken by the players to solve the task. This time is given by parameter τ_k. Therefore, given a fixed number of players $\bar{k}$, the problem of putting out the fire is trivially solved by summing $\tau_{\bar{k}}$ to the objective function value.

4.2 Moving the Players from the Fire Extinguishers to the Fire

The model assumes that a player equipped with a fire extinguisher cannot be harmed by fire. As a consequence, at this step we only care about the time taken by the player to get to destination. Given $\bar{v}_e$ and $\bar{v}_f$, the locations of a determined extinguisher and of the fire, respectively, this problem can be solved as a Shortest Path Problem (SPP) on graph G. In graph theory, the SPP is the problem of finding a path between two vertices (or nodes) in a graph such that the sum of the weights of its constituent edges is minimized. For its solution we used a labelling algorithm such as the Dijkstra's algorithm [6].

4.3 Moving the Players to the Fire Extinguishers

Differently from the previous subproblem, now the damage taken by the has to be taken into account and the objective is to minimize both the time taken to get to destination and the damage sustained by the player along the path. Given $\bar{v}_p$ and $\bar{v}_e$, the locations of a determined player and of a determined fire extinguisher, respectively, the resulting problem is a Multiobjective Shortest Path Problem (MSPP) on graph G. The MSPP was originally proposed by Vincke [7] and its objective is to to determine a path that minimizes simultaneously all the criteria under consideration. Usually, there is not a path exhibiting the best value for all the criteria. Hence, for the MSPP, we are looking for the Pareto Frontier (PF), i.e., the set of non-dominated paths in the network, that is, paths for which it is not possible to find a better value for one criterion without getting worse for some of the other criteria. The MSPP is known as being hard to solve and the large number of applications of the MSPP led to the development of various strategies to obtain the optimal solutions. In this work, we solved the MSPP by using a labelling algorithm that provides PF_{pe}, i.e., the Pareto Frontier comprised of all the paths from $\bar{v}_p$ to $\bar{v}_e$ that (1) minimize the time taken and (2) minimize the damage sustained by the player p to get to the fire extinguisher e location. From PF_{pe} we obtain $\bar{PF}_{pe}$, which is the set of feasible non-dominated paths, i.e., the paths having a total damage that is less than or equal to D.

4.4 Assigning the Fire Extinguisher to the Players

Once we solved the previous two subproblems, we can compute ρ_{pef}, the best paths to get a player p to the fire when assigned to the fire extinguisher e:

$$\rho_{pef} = \arg\min \{time\,(\rho_{pe})\} + \rho_{ef} \tag{20}$$

where $\rho_{pe} \in \bar{PF}_{pe}$ is a feasible non-dominated path from p to e, $time\,(\rho_{pe})$ is the value of the time attribute of path ρ_{pe}, ρ_{ef} is the path from e to the fire, and "+" represent the path-merging operator. Once we have all paths ρ_{pef} and their time attribute $time_{pef}$, we can compute the best way of assigning players to fire extinguishers.

Now, let $\bar{k}$ be the fixed number of players participating in the task and w_{pe} be binary variables taking value 1 if the player p is assigned to the fire extinguisher e, and 0 otherwise. The problem of determining the best assignment of fire extinguishers to players can be formulated as:

$$\min \quad \max \ time_{pef} w_{pe} \tag{21}$$

$$s.t. \quad \textstyle\sum_{e\in E} w_{pe} \leq 1 \qquad \forall p \in P \tag{22}$$

$$\textstyle\sum_{p\in P} w_{pe} \leq 1 \qquad \forall e \in E \tag{23}$$

$$\textstyle\sum_{p\in P}\sum_{e\in E} w_{pe} = \bar{k} \tag{24}$$

$$w_{pe} \in \{0,1\} \tag{25}$$

The objective is to minimize the worst time. Constraints (22) and (23) impose that a player can have at most one fire extinguisher assigned, and that a fire extinguisher can be assigned to one player at most. Finally, constraint (24) imposes that the number of participating players must be exactly $\bar{k}$. This problem is a generalized assignment problem with min-max objective function [8].

For its solution we propose a bounded enumeration algorithm that solves the problem for all the values of $\bar{k} = 1, \ldots, K$ at the same time, since the solution space of the problem for a lower value of $\bar{k}$ is entirely contained in the solution space of the problem for a higher value of $\bar{k}$. The algorithm features a bound that excludes partial solutions that cannot lead to any optimal solution, reducing the computational complexity of the problem significantly.

4.5 Deciding how many Players take part in the Task

The enumeration algorithm for the assignment problem defines the best values for variables w^k_{pe}, i.e., the assignment variables for the case with k players. The best number of players to be involved in the task can be determined as:

$$k^\star = \arg\min_{k=1,\ldots,K} \left\{ \max_{p\in P} \{time_{pef} w^k_{pe}\} + \tau_k \right\} \tag{26}$$

4.6 Reconstructing the Optimal Solution

Once we know $k^\star$, the best assignment of fire extinguishers to players is given by $w_{pe}^{k^\star}$. Also, the paths to be taken by the players are given by ρ_{pef} and the total time taken by the group to put out the fire is $\max_{p \in P} \left\{ time_{pef} w_{pe}^{k} \right\} + \tau_k$.

5 Computational Experience

5.1 Dataset

5.2 Experiments

6 Conclusions

References

[1] I. Althöfer, Computer-aided game inventing, Technical Report, Friedrich Schiller Universität Jena, 2003 [Online].

[2] H. Iida, K. Takahara, J. Nagashima, Y. Kajihara and T. Hashimoto, An application of game-refinement theory to Mah Jong, ICEC 2004 (LNCS 3116), M. Rauterberg Ed. Berlin: Springer, 445-450, 2004.

[3] C. Browne, Connection Games: Variations on a Theme, Natick, Massachussetts: AK Peters, 2005.

[4] C. Browne and F. Maire, Evolutionary Game Design, IEEE Transactions on Computational Intelligence and AI in Games 2(1):1-16, 2010.

[5] General Algebraic Modeling System (GAMS), GAMS Development Corporation, http://www.gams.com.

[6] E. W. Dijkstra, A note on two problems in connexion with graphs, Numerische Mathematik 1:269-271, 1959.

[7] P. Vincke, Problèmes multicritères, Cahiers du Centre d'Etudes de Recherche Opérationelle, 16, 425-439, 1974.

[8] S. Martello, P. Toth, Generalized assignment problems, in Algorithms and Computation, Lecture Notes in Computer Science, 650:351-369, 1992.