Metaheuristic Entry Points for Harnessing Human Computation in Mainstream Games

Peter Jamieson, Lindsay Grace, Jack Hall, and Aditya Wibowo

Miami University
Oxford, OH
jamiespa@muohio.edu

Abstract. In this work, we describe a promising approach to harnessing human computation in mainstream video games. Our hypothesis is that one of the best approaches to seamlessly incorporating harnessing within these games is by examining existing game mechanics and matching them to meta-heuristic algorithms. In particular, we believe that the best choices for early exploration of this problem are nature inspired metaheuristic algorithms for combinatorial optimization problems. In this paper, we will describe the problem in more detail and describe two proof of concept games that demonstrate the viability of this approach. The first game is designed to be incorporated in Real-time Strategy games within the resource gathering aspects of these games, and the algorithm and problem that are used is related to Ant Colony Optimization and the Traveling Salesman Problem. The second game explores a racing game where the problem and algorithm are embedded in the numerical characteristics of the racer such as speed, agility, and jump power. These characteristics represent current solutions to different traveling salesman problems, and the solutions are modified through training and mating of racers; this is analogous to mutations and crossbreeding in genetic algorithms.

1 Introduction

Hybrid computation systems that can take advantage of computation (or problem solving) skills of both humans and computation machines have the potential to solve complex real-world problems that at present may be tedious human tasks, challenging unsolved problems, or are complex tasks that no computer algorithm has yet been created to solve. In the case of humans, we are able to make decisions based on observed patterns and understanding of the big picture, but we tire from monotonous tasks and are slow when dealing with low-level computational tasks. Computing machines, on the other hand, are tireless and accurate, but are hard to program/design to process high-level concepts. The challenge is merging these two entities so that humans enjoy the computation activity as a game, and the overall system provides better solutions to the real-world problems compared to an algorithm or human alone.

Ultimately, if we could harness human computation during the play of games, then we would recycle a massive amount of thought that, currently, is directed

A.A. Ozok and P. Zaphiris (Eds.): OCSC/HCII 2013, LNCS 8029, pp. 156–163, 2013.

purely to entertainment. With the linking of people via the web, we have a human cloud that many have tried to harness for a number of tasks. Popular crowd sourcing examples, such as Amazon's Mechanical Turk (MTurk), and human computing games (HCG [2]) such as Foldit [7] have shown some success in harnessing humans for tasks. Still, mainstream video games (including casual games) are played for at least an hour a week by 135 million people in the U.S.A. according to Macchiarella's white paper report [14]. That is a massive "thought" resource that to this day remains untapped.

In this work, we describe an approach to harnessing human computation in mainstream video games. Our hypothesis is that early approaches to incorporating harnessing withing games will happen by examining existing game mechanics and matching them to meta-heuristic algorithms. In particular, we believe that the best choices for early exploration of this problem are nature inspired meta-heuristic algorithms for combinatorial optimization problems.

To test our hypothesis we, first, explain the challenges associated with including HCGs in mainstream video games. Specifically, we identify how creating isomorphs of problems for games and puzzles is particularly challenging.

We then describe two proof-of-concept games that demonstrate the viability of our approach. The first game is designed to be incorporated in Real-time Strategy (RTS) games within the resource gathering aspects of these games. The algorithm and problem that are used as the HCG connection is related to Ant Colony Optimization (ACO) [4] and the Traveling Salesman Problem (TSP). The second game explores a racing game where the problem and algorithm are embedded in the numerical characteristics of the racer such as speed, agility, and jump power. These characteristics represent current solutions to different TSPs, and the solutions to these problems are modified through training and mating of racers; this is analogous to mutations and crossbreeding in genetic algorithms (GAs) [9].

2 Background and Definitions

The Human Cloud (HC) can be described as a network of intelligence, connected through existing systems like the Internet. Harnessing the HC has been done in a number of projects in a form of crowd-sourcing, which is defined in this paper as using the knowledge and effort of a large group of individuals. In particular, Human Computing Games (HCGs) have been used to harness the HC and include games such as protein folding [7], picture identification [5], and other types of human computation have been achieved without the game aspect such as security camera searches [10] and searching for space dust [19].

HCG's and other types of productive play fall under the greater domain of games with a purpose (GWAP) (defined by von Ahn) and the general and contested term serious games. Since von Ahn's original work on a ESP [5], over 40 HCGs have been presented at conferences and in journals by a growing number of researchers.

One of these games, Foldit has had tremendous success [7], it is a game developed to allow players to interact with complex proteins, attempting to solve one of the problems facing biological science - how to predict protein structure. Foldit presents 3D abstraction of proteins and simplifies the chemistry behind protein folding by making the problem into a game. It is clear to the user that they are folding proteins as no abstraction is made, but users are rewarded for their efforts by a driving goal to get to the top of the leader board. Foldit develops solutions that are better than those produced by machines by encouraging competition between the 50,000+ players of the game and has been successful.

Most HCGs are projects with real-world problems and these problems are presented in what we call the "direct form". Users are aware of the real-world problems since the game is presented in this manner, and the players are motivated to help by competing against others for status or other direct incentives such as money or recognition. The limitation of this approach is that only a specific problem can be solved in the direct approach, the game must be designed from the real-world problem, and the larger potential population of game players who just play for entertainment tends to be left untapped.

2.1 Isomorphs

Humans are able to make heuristic problem solving decisions based on observed patterns and an understanding of the big picture, but we tend to tire with monotonous tasks that are not motivated by intrinsic rewards, and we are slow when dealing with low-level computational tasks. One of the major challenges in incorporating HCG aspects into mainstream video games is how can we take a real-world computation problem and map it into the game. More, importantly, can this mapping be done in a way so that the problem being solved just seems part of the game.

To achieve this, we must address what is called an isomorph. More specifically, an isomorphic problem has multiple presentation formats at the surface level, but are the same problem underneath. For example, a problem like mowing the lawn requires that a mower be controlled to both efficiently cover the area that needs to be mowed and avoid any live obstacles that might enter the mowers path such as children or animals. An isomorph of the lawn mowing problems could be a coloring game where the goal is to draw in an area without lifting the crayon and avoiding random targets that might try to make you lift your crayon. A similar term to isomorphing in the field of computer science is reduction.

Isomorphic problems have been of interest to cognitive psychologists, and they have been used to help us understand strategical approaches people take to solving problems [18], [13]. An equivalent problem example that illustrates the basic concept of an isomorph is for tic-tac-toe. Zhang *et. al.* [22] show some common ways of presenting tic-tac-toe, and we have replicated one of these in figure 1. In figure 1, the number game is shown where players take turns at picking a number (by coloring in a circle) with the goal of picking three numbers to total fifteen exactly. This game is the same as tic-tac-toe, and the figure shows how these numbers can map to locations on the tic-tac-toe board.

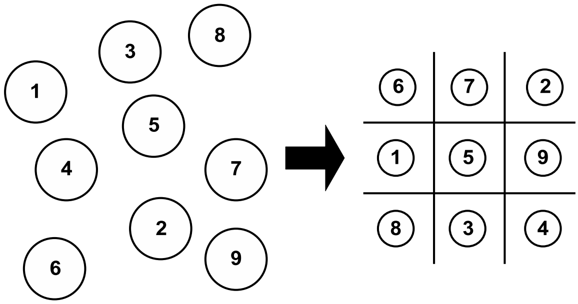

Fig. 1. How a number game of picking 3 numbers that make 15 is an isomorph of tic-tac-toe board

2.2 Meta-heuristics Algorithms and Combinatorial Problems

Meta-heuristic algorithms are one type of algorithm used to solve combinatorial optimization problems [15]. An excellent survey of some of these algorithms is described by Blum *et. al.* [1] and these algorithms can be considered as guiding a search through a search space with iterative improvements. Common meta-heuristic algorithms include algorithms such as Simulated Annealing [12], Tabu Search [8], Ant Colony Optimizations [4], Genetic Algorithms [9], and newer ones such as Firefly [21] and Bees algorithms [16].

Blum *et. al.* [1] try to classify many of these algorithms and use various categories. Of particular interest to this project is those that are classified as nature inspired versus non-nature inspired. We believe that the nature inspired algorithms may be easier to use as interfaces for good isomorphs. We think this may be true based on the connection between human actions in nature, game enjoyment and its relationship to our basic needs as described by Falstein in his game design article, "Natural Funativity" [6].

Regardless of the classification, we hypothesize that meta-heuristic algorithms are an excellent interface point to think of good game isomorphs that humans can work with computers to solve real-world problems. Meta-heuristics have been used to solve a number of classical problems such as the traveling salesman problem, the quadratic assignment problem, and the timetabling and scheduling problems. In terms of a tighter connection to real-world problems there are too many applications to list here. Recently, at CEC 2011 a report was prepared that lays out 15 problems to help evaluate evolutionary algorithms [3], and these problems are in areas such as energy and power distribution, chemical production, economics, antenna design, and aerospace.

The no-free-lunch theorem [20], suggests that there does not exist an algorithm for solving all optimization problems that is generally on average better than competitors. This means that choosing one algorithm over another is not that important, but since our algorithmic based games will be a combination of human and computer, we hypothesize that this theorem is still true, but the algorithms aspects, which include how visualizations of the cost function are presented to the gamer, will have an impact on the quality of solutions.

Finally, some meta-heuristic algorithms are classified as nature inspired versus non-nature inspired [1]. We hypothesize that these nature inspired algorithms may be easier to use as interfaces for good isomorphs since there is a connection between human actions in nature, game enjoyment and its relationship to our basic needs as described by Falstein in his game design article, "Natural Funativity" [6].

3 Proof-of-Concept Games

In this section, we describe two games that we have developed to study how well nature inspired meta-heuristic games fit within aspects of mainstream video games. The section following this one will provide discussion on what we have learned from these experiences.

3.1 Swarm-Miner for ACO in Games

Swarm-miner is a game that is inspired by the resource collection in RTS games. The basic idea of our game is to have the human interact with resource collectors in a game to improve the quantity of the collected resources (as reflected by minimizing the cost function of the TSP).

The Ant Colony Optimization (ACO) algorithm is utilized for solving the traveling salesman problem in this game. Without human assistance, the ACO algorithm solves the TSP problem by simulating agents, ants, that investigate paths with the simple heuristic that edges with higher pheromone levels are more likely to lead to improved paths and thus are more likely to be chosen by ants. As many ants investigate paths, the pheromone levels on those edges that correspond to a path segment that is shortest are increased.

The objective of our Swarm-Miner game is to mix the solution of shortest path with ACO and humans where the game player wants to find the shortest path for as many maps as possible, thus improving the resource collection. Each map represents, in theory, one of the areas that a resource collector in an RTS game would be harvesting resources. In the proof-of-concept game, rank is determined by those users who have found the current best solution to one of several puzzles hosted on a server. The game is intended to draw on human intuition in order to assist the ACO algorithm in solving this difficult problem. Users utilize various mechanics for altering the paths that will be searched and the likelihood that a path edge will be incorporated into the algorithm's search. Two of these mechanics include doping and re-routing. The doping mechanic allows users to change the pheromone level on an edge that is currently in the ACO optimal path. The re-routing mechanic allows users to re-order the sequence of nodes in which a continuous section of the optimal path follows.

While ACO is one of the most powerful algorithms that can be used to solve the TSP, we have observed during the development process that user solutions are usually equally or more optimal than those solutions that were found by the algorithm computing idly (i.e. without human assistance).

3.2 Monster Racer for GA in Games

Monster Racer is a third-person perspective video game where players control a monster that competes in races similar in nature to Mario Cart. In addition to racing, players interact with their monsters by training them to improve the monster's racing attributes, which includes speed, agility and jump power. This can be done by either training or breeding monsters which translates to steps within a Genetic Algorithm (GA) operations, mutation and crossover, respectively.

In the beginning of the game, a player's monster starts with certain attributes generated by the game. These attributes are actually directly related to the cost function of a solution to an instance of a Traveling Salesman Problem (TSP). For example, a specific TSP problem for speed is shared by all game players, and the initial speed for a player's monster is created by running a GA for a certain number of generations with a unique random seed. Therefore, for this game, three different GAs are run at the beginning, and each produces a solution to three distinct TSPs. The cost function for these solutions is then used to calculate the monsters attributes - agility, speed, and jumping.

As players compete in races, points are earned, which then can be used to either train or breed a player's monster. Training performs random mutations on the current solution for an attribute selected to be trained. If training is successful, in the back-end, the GA has found a better solution for the TSP problem, and translates this solution as an improved monster attribute.

Breeding can also be done to improve a player's monster. Players use allocated points to look at other player's monsters, and then the player chooses one of the other monsters to breed with. Breeding is actually a series of crossover operations using the genome of both monster's attributes. If any of the crossover operations produce a better solution, then the monsters attribute is increased accordingly.

As the game progresses, players compete in a traditional racing game, and at the same time, through competition try to improve the abilities of their monsters in a similar way that a GA behaves.

4 Discussion on Our Approach

Developing these two games has taught us a few important lessons about both how to incorporate meta-heuristics as an entry point for HCGs in mainstream video games and some ideas on how this field of research might move forward.

First, *the isomorph of meta-heuristics algorithms into games is a reasonable match.* The choice of GA and ACO are algorithms that fit well with traditional game mechanics. Though the isomorphs are not seamlessly integrated into games, they match well with game play. For example, Swarm-Miner replicates the resource gathering stages in a classical RTS, but no modern game requires micromanaging resource gathering. Similarly, monster racer incorporates the GA through avatar characteristics, and the cross-breeding and mutation are mechanics within the game, but the cross-breeding mechanic, in particular, is a rare mechanic in mainstream video games.

Second, *the granularity of the HCG is small in comparison to how it fits in the entire mainstream game.* This means that HCGs in mainstream video games will, likely, only be a subset of the entire game. This allows designers to consider sub-problems of the video game as harnessing points for mini-HCGs. For example, swarm-miner represents only the resource mining stage of a much larger RTS game.

Third, *the creation of HCGs and crowd sourcing systems lacks a framework to aid in the development, aggregation, and quality control of such solutions.* Both of these games took a significant time to create even in their proof-of-concept form. There appears to be a major opportunity to create a crowd sourcing framework that would allow researchers to quickly design their ideas and experiment with them. In particular, as these systems develop there are three issues that are interesting as research questions, but require significant infrastructure development.

– User action tracking: as we attempt to understand how hybrid systems solve problems differently from traditional computational methods there is a need for user behavior tracking. For example, Foldit observed users and created new and more efficient automated algorithms based on their solutions [11]
– Aggregation: as identified in Quinn *et. al.* [17], there is a need to collect answers into a global solution. The question remains, what are the most efficient methods to do this as these systems scale in terms of users?
– Quality control: also identified in Quinn *et. al.* [17], there is a need to regulate the solutions provided by users to protect the system from poor and bad behavior.

To test even one solution in this space is very difficult, and a framework that speeds up this development would provide a significant step forward in this area of research.

5 Conclusion

In this paper, we discussed the benefit of creating HCGs defined by meta-heuristic algorithms to harness human computation in mainstream video games. In particular, we described two proof-of-concept games we have developed that test out our ideas. From these experiences we have learned a few important lessons, and describe how a framework for crowd sourcing applications (both games and non-games) would benefit future research in this domain.

References

1. Blum, C., Roli, A.: Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Comput. Surv. 35, 268–308 (2003), `http://doi.acm.org/10.1145/937503.937505`
2. Cusack, C., Largent, J., Alfuth, R., Klask, K.: Online games as social-computational systems for solving np-complete problems. In: Meaningful Play (2010)

3. Das, S., Suganthan, P.: Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems. Tech. rep. (2010), http://web.mysites.ntu.edu.sg/epnsugan/PublicSite/SharedDocuments/CEC2011-RWP/Tech-Rep.pdf
4. Dorigo, M., Maniezzo, V., Colorni, A.: Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 26(1), 29–41 (1996), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=484436&tag=1
5. ESP Game: (2008), http://www.gwap.com/gwap/gamesPreview/espgame/
6. Falstein, N.: Natural Funativity. Gamasutra.com (2004), http://www.gamasutra.com/features/20041110/falstein_pfv.html
7. FoldIt: (2008) http://fold.it/portal/
8. Glover, F.: Future paths for integer programming and links to artificial intelligence. Comput. Oper. Res. 13, 533–549 (1986), http://dl.acm.org/citation.cfm?id=15310.15311
9. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning, 1st edn. Addison-Wesley Professional (January 1989), http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0201157675
10. Internet Eyes: (2009), http://interneteyes.co.uk/
11. Khatib, F., Cooper, S., Tyka, M.D., Xu, K., Makedon, I., Popovic, Z., Baker, D., Players, F.: Algorithm discovery by protein folding game players. Proceedings of the National Academy of Sciences (2011), http://www.ts-si.org/files/doi101073pnas1115898108.pdf
12. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by Simulated Annealing. Science 220(4598), 671–680 (1983)
13. Kotovsky, K., Hayes, J.R., Simon, H.A.: Tower of Hanoi - Problem Isomorphs and Solution Processes. Bulletin of the Psychonomic Society 22(4), 290 (1984)
14. Macchiarella, P.: Trends in Digital Gaming: Free-to-Play, Social, and Mobile Games. Tech. rep. (2012)
15. Papadimitriou, C.H., Steiglitz, K.: Optimization Algorithms and Complexity. Dover Publications, Inc., New York (1982)
16. Pham, D., Ghanbarzadeh, A., Koç, E., Otri, S., Rahim, S., Zaidi, M.: The bees algorithm, a novel tool for complex optimisation problems. In: Virtual International Conference on Intelligent Production Machines and Systems, pp. 454–459 (2006), http://www.bees-algorithm.com/modules/2/4.pdf
17. Quinn, A.J., Bederson, B.B.: Human computation: a survey and taxonomy of a growing field. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1403–1412 (2011), http://doi.acm.org/10.1145/1978942.1979148
18. Simon, H.A., Hayes, J.R.: Understanding Process - Problem Isomorphs. Cognitive Psychology 8(2), 165–190 (1976)
19. Stradust@Home: (2009), http://stardustathome.ssl.berkeley.edu/
20. Wolpert, D.H., Macready, W.G.: No Free Lunch Theorems for Search, REPORT SFI-TR-95-02-010. Tech. rep. (1996)
21. Yang, X.S.: Nature-Inspired Metaheuristic Algorithms. Luniver Press (2008)
22. Zhang, J., Johnson, T., Wang, H.: Isomorphic Representations Lead to the Discovery of Different Forms of a Common Strategy with Different Degrees of Generality. In: Proceedings of the 20th Annual Conference of the Cognitive Science Society (1998), http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=93CB1A5E27B562C8B7BD109A1BF3A241?doi=10.1.1.139.9596&rep=rep1&type=pdf