On Validity of Personalized Planning and Integrated Testing Algorithms in Reproducible Games

Sedrak Grigoryan
Division of Computational and
Cognitive networks, IPIA, NAS RA
Yerevan, Armenia
e-mail: addressforsd@gmail.com

ABSTRACT
We describe steps and means required to present and solve problems of the class of Reproducible Game Trees by means of the package RGT Solver. Namely, given specification of RGT problems and perspective strategy related knowledge RGT Solver by PPIT algorithms elaborates and tests plans, selects the best actions for the given plan and processes them.
We prove the validity of PPIT by successful experiments in elaboration of endgame strategies for chess RGT model problem and provide an evidence of validity of PPIT for known Retie etude.

Keywords
Competition, strategy, planning, chess

1. INTRODUCTION
1.1 We are looking for general methods of solution of problems presented as games with Reproducible Game Trees (RGT) [1], namely, problems that meet the following requirements:
- there are (a) interacting actors (players, competitors, etc.) performing (b) identified types of actions at (c) specified moments of time and in (d) specified types of situations
- the actors have identified benefits
- the situations the actors act in and transform after the actions can be specified by certain rules, regularities.

RGT class includes, particularly, the problems of protection of networks from intrusions, management in oligopoly competitions, computational anomalies detection and correction and chess-like combinatorial problems.
1.2 Our team is developing RGT Solver [2] for searching optimal strategies for RGT problems. Solver is a package aimed to acquire strategic expert knowledge to become comparable with human in solving competing and combating problems. Regular improvement of Solver by expert knowledge we study, particularly, for chess RGT model problem, where knowledge representation and its consistent inclusion into chess programs stay central since the pioneering work by Shannon in 1950.
1.3 Approaches of inclusion of chess knowledge into strategy formation process are described in [4].
Previously suggested PPIT (Personalized Planning and Integrated Testing) [1, 3] algorithms search for optimal strategies by usage of expert knowledge. The algorithms had been tested for a variety of problems, such as Retie and Nodarishvili chess etudes [3], intrusion protection problems [5].
In [6] planning-based strategy searching algorithms within the frame of Solver package are described, where plans are certain general descriptions of strategies. Occupying the center or the corners of the board could be plans for one to play with. Each plan represents a hierarchy of the goals which a player tries to achieve in current situation while playing by the plan. The essence of the plans is to select goals which get the maximal profit.
The PPIT program was designed as a composition of the following basic units:
Reducing Hopeless Plans (RHP)
Choosing Plans with Max Utility (CPMU)
Generating Moves by a Plan (GMP)
1.4 The current state of package lets us embed specified RGT games, related expert knowledge and searches beneficial strategies according to the given plans.
Solver lets us select the best plan from given plans in the situation and search strategies for the selected plan to play with.
In the paper we present the chain of bringing RGT problems into Solver and solving them for chess.
Given specification of RGT problems and strategy related perspective knowledge, the planning algorithm returns the list of best actions selected for the given situation.
We provide evidence on validity of PPIT based knowledge based planning by experiments in chess endgames and known Retie etude.
These findings let us integrate PPIT into Solver and prepare grounds to extend the process of plans execution by revealing plans from given game-related knowledge and choosing the best plan for the given situation.

2. EMBEDDING RGT PROBLEMS AND STRATEGY RELATED KNOWLEDGE INTO SOLVER
Bringing the RGT problem into the RGT Solver for the expert from the given problem specification requires several

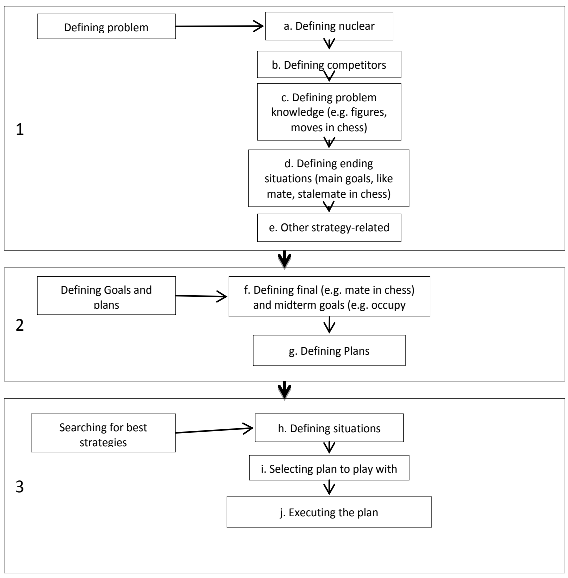

Fig. 1 Solver in embedding and solving problems.

steps to perform. In the following chapters we provide detailed description of how problems, strategy related knowledge and situations can be integrated in RGT Solver then processed for obtaining the list of actions realizing plans.

2.1. EMBEDDING RGT PROBLEMS

Integration of problems into Solver is started from identification of the level of nuclear units. Generally, RGT problems can be specified in variety of levels of abstraction.
We present problems by playing actors, the actions of the actors, situations and rules transforming situations after the actions. Thus, chess presentation in Solver starts from presenting chess boards as a composition of X and Y coordinates of fields, the pieces including their colors and types, IDs of players, i.e., the competing sides, and all possible moves of the pieces.
As it was discussed in [7] 4 nucleus abstracts are identified for chess - FigureColor, FigureType, cordX, cordY. Each field on the board is a composition of these 4 abstracts, and each figure is inherited from the field abstract by identifying FigureColor!=0, FigureType!=0. Legal moves of all pieces are needed as well. The abstract Move itself is a virtual abstract and each particular type of moves, e.g. MoveByKnight, are inherited from that Move virtual abstract.
Since MoveByKnight also needs to be specified there will be 8 types of moves MoveByKnight1-MoveByKnight8. For finishing the embedding of problems, the abstracts mate, stalemate and draw in chess have to be determined.
Detailed descriptions of those abstracts are given in [8].
Chess goals are a type of expert knowledge. For chess the main goal is mate and, generally, is sufficient to play exhaustive chess. To be more efficient Solver provides an ability to define strategy related knowledge. In [1, 3, 5] evidence was provided in essential enhancement of the efficiency by embedding strategy knowledge into Solver.

2.2. PLANNING IN SOLVER

Earlier PPIT algorithms [6] were using injected strategy related knowledge and did not provide regular means for that.
Here we add functions for inference and processing plans from given goals while they have to be embedded by experts. Then the validity of interface of PPIT for chess knowledge is demonstrated.

2.2.1. Interface to input situations, goals and plans

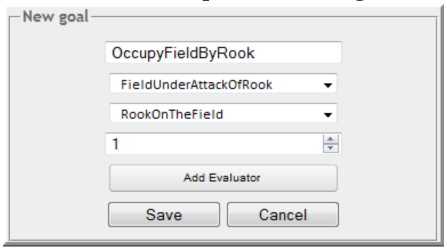

Fig. 2 Interface to define goals

As it is shown in Fig. 2 the following information have to be specified: a) precondition and postcondition of goals to be satisfied in situations before and after achieving the goals, b) max depth of game trees to be generated to achieve goals and c) criteria for evaluation of achievement of the goals. Whether the goals are final or intermediate have to indicated to mark ending situations of the problem, say, mate, stalemate or other ending positions in chess.
Plans are goals chained by their priorities. In the interface users need to add the goal and indicate its priority in the plan they belong to or the list of existing ones. Plans can include only one goal, e.g., the final one, or can include several goals in solving given sub-problems.

The structure of situations described in [9] consists of different nucleus types and IDs uniting them in groups. The interface, at present was tested for chess situations while the expansion of its functions to situations of other RGT problems needed.

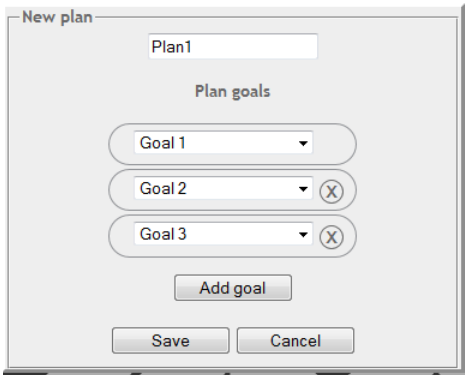

Fig. 3 Interface to define goals

This is caused by own sets of nucleus types of RGT problems making impossible development of unified interface for processing situations of all types.

2.2.2. Searching strategies by plans

For RGT problems we input situations by the interface. Thus for chess we define situations on the board, insert plans and a request for their processing. The processing algorithm returns the list of recommended best actions for the plans. After applying each action to the situation another attempts of processing of plans in new situations are expected, until the plan is achieved or its achievement becomes impossible. The structures of plans and goals as well as the algorithms of strategy searching by plans are detailed in [6].

2.3. CONCLUDING CHAPTER 2

RGT Solver is a package to search RGT strategies using specification of RGT problems and knowledge related to strategies.
Interface to define goals and plans is developed. Input of RGT problems, knowledge on their situations, goals and plans are demonstrated. The realized version of Strategy search PPIT algorithm is described.

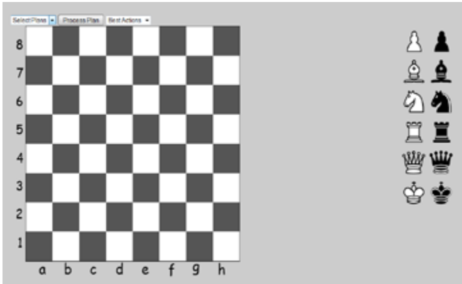
Fig. 4 Chess interface

3. CHESS VALIDATION OF PPIT ALGORITHM

The integration of some RGT problems into Solver was demonstrated, particularly, in [10, 7] and successful work of versions of PPIT in [6].
Here we are focusing on planning by PPIT and demonstrate its effectiveness for known chess Retie etude, particularly we are bringing the knowledge inserted into Solver, goals and the plan of reaching the expected result, plan processing for the etude. The expected result for the etude is achieving a drawing situation for white.

3.1. CHESS REGULARITIES, GOALS AND PLANS IN SOLVER

For Retie etude solution, first of all, chess rules have to be inserted.
As it was mentioned above chess is initially defined by 4 abstracts including colors of pieces, types of pieces, their coordinate, etc.

Fig. 5 Nuclear units in chess definition

Knight and MoveKnight chess domain knowledge pieces are presented.

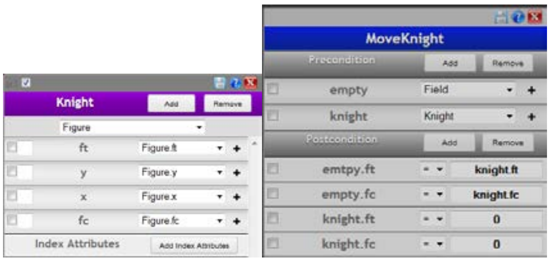

Fig. 6 Defining chess rules

In [3] a winning plan for Retie etude with usage of strategy knowledge and its sequel strategy searching result was given. Here we are deepening it by providing means for regular knowledge acquisition and more independent from human search of Retie solution. They include certain structures, interface and Retie etude planning algorithm.

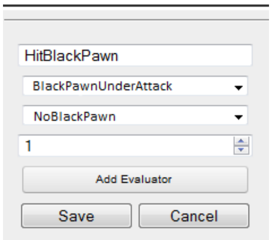

Fig. 7 HitBlackPawn goal

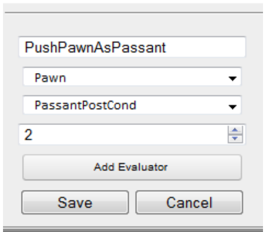

Fig. 8 PushPawnAsPassant goal

The winning plan given in [3] is a chain of instructions like the following ones: 1. Hit opponent pawn, 2. Pass the pawn, 3. Protect own pawn. 4 Stay maximally close to pawns. 1 – 4 are goals, which are being embedded into the Solver too.

HitPawn goal consist of precondition fieldUnderAttack, postcondition NoBlackPawn, and the depth is 1.

PushPawnAsPassant goal postcondition is an abstract consisting of two abstracts PawnPushed, which defines that Pawn coordinate is changed from postcondition situation and PawnNotAttacked abstract is defined as a

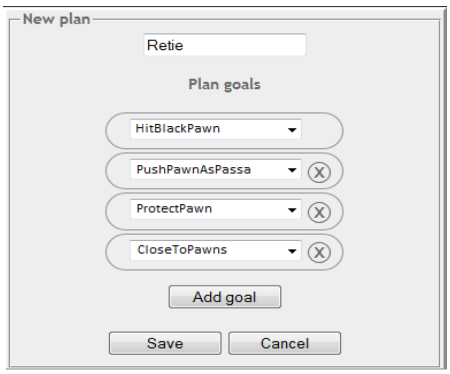

Fig. 9 Retie etude plan

virtual abstract having two specifications PawnNotUnderAttack and PawnIsProtected.

Similarly other two goals of the given plan are defined. ProtectPawn goal with precondition of Pawn on the board, postcondition of PawnIsProtected, with depth 1 and maxixmal value of "king.y" criterion. CloseToPawns goal with precondition and postcondition of any situations and evaluator indicating that the distance between the king and two pawns is minimal.

Retie Plan is defined as a composition of these 4 goals in priorities.

3.2. PPIT RESOLUTION OF RETIE ETUDE

In 3.1 plan of Retie etude is defined in the Solver. Here we are presenting resulted processing of the plan.

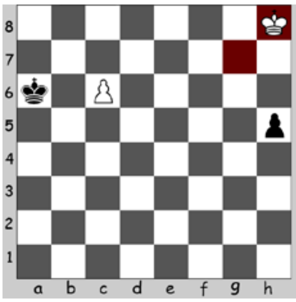

Fig. 10 Initial situation and first suggested action

On the given situation Retie plan is selected and processing of the situation is requested.

Here we are going to show how the strategy search is done for a certain chain of moves done by the Solver for Retie etude. The processing algorithm [6] works as follows. For the initial situation the algorithm searches for moves for achieving HitBlackPawn goal. In the given situation the situation does not match the precondition of the goal and thus, goal can't be achieved. The process passes to next priority goal, PushPawnAsPassant. For this goal precondition is matching to the situation, so the algorithm generates a game tree with depth of 2, i.e., white and black moves. For all of white moves black has moves that bring to a situation where postcondition of the goal is not matched, after c7 move Kb7 answer by black can be considered which does not satisfy postcondition of the goal, so this goal does not give any action to perform, too. Processing takes next goal ProtectPawn goal. Since there is no situation where pawn is protected, which means no move satisfies postcondition of this goal, too. The last goal to process by the algorithm is CloseToPawns goal which has no precondition and postcondition and for each of moves it checks for distances between king and two pawns. From all of possible moves by white best value of criterion for minimal distance between king and two pawns is calculated for Kg7. So Solver suggests move Kg7.

Next we will just bring a certain game by black and suggested by the Solver strategy accordingly. After move Kg7, let's assume black plays h4.

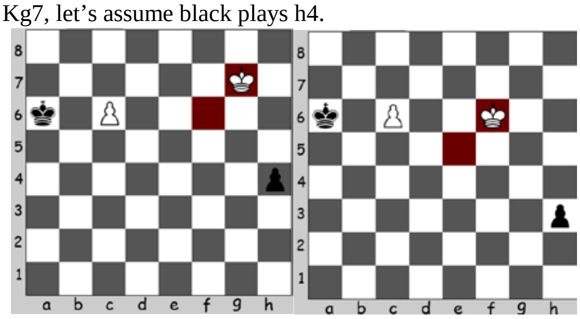

Fig. 11 Second suggested action

Situation after h4 move is shown on the left image of Fig. 11. The given situation is processed in Solver similar to the initial situation and Kf6 move is suggested.

Next black h3 move is considered and situation after that move is on the right image of Fig. 11. Processing is similar to previous two steps and accordingly Ke5 move is suggested.

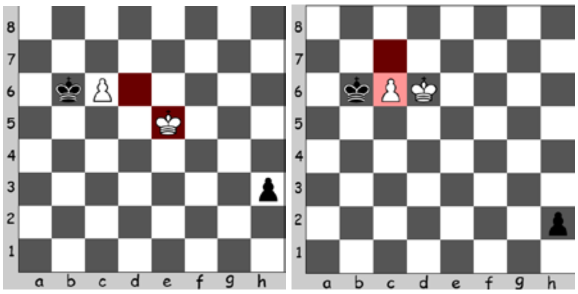

Fig. 12 Next suggested moves

Let's assume now black plays Kb6 (Fig. 12, left image). HitBlackPawn and PushPawnAsPassant goals are still not achievable. ProtectPawn goal processing finds two moves satistying the postcondition – Kd5 and Kd6. Processing by the evaluator of the goal Kd6 is selected because critrerion "king.y" has maximal value for Kd6 move.

After Kd6 let's assume black plays h2 (Fig. 12, right image). HitBlackPawn goal is not achievable on this situation too. PushPawnAsPassant goal is achievable and c7 is suggested by Solver.

So the game was 1. Kg7, h4 2. Kf6, h3 3. Ke5 Kb6 4. Kd6 h2 5. c7… 1.Kg7, 2. Kf6, 3. Ke5 moves are selected by CloseToPawns goal. 4. Ke6 is selected by ProtectPawn goal which has higher priority and 5. c7 move is selected by PushPawnAsPassant goal.

As it is demonstrated Solver strategy for Retie etude brought to a drawing situation, which is the expected result.

3.3. CONCLUDING CHAPTER 3

Chess specification and strategy related plan for Retie etude are provided and its solution by PPIT is demonstrated.

4. CONCLUSION

a. RGT Solver searches for optimal strategies in RGT games by the problem specification and strategy related knowledge. The ability of bringing the RGT problem from the given specification as well as strategy related knowledge integration into the Solver in a regular manner is demonstrated.

b. Plans and goals integration interface is developed according to previously designed structures. Situation input interface is discussed, where chess situation interface is already integrated and other RGT problems require their own situation interface development and integration. Strategy searching by the given plan is described.

c. Developed planning interface and overall RGT Solver package validities tested and proved for chess well known Retie etude.

d. Our ongoing efforts are doing the interface more flexibility to let Solver generate and process plans with minimal intervention of the users.

5. ACKNOWLEDGEMENTS

Author expresses his deep gratitude to Professor Edward Pogossian for huge input in the work and Narek Ektubaryan for support in development of Planning interface development in Solver.

References

[1] E. Pogossian, V. Vahradyan and A. Grigoryan, "On competing agents consistent with expert knowledge", *Lecture Notes in Computer Science, AIS-ADM-07: The International Workshop on Autonomous Intelligent Systems - Agents and Data Mining*, pp. 229-241, St. Petersburg, Russia, June 6-7, 2007.

[2] K. Khachatryan and S. Grigoryan, "Java programs for presentation and acquisition of meanings in SSRGT games", *Proceedings of SEUA Annual conference*, pp. 127-135, Yerevan, Armenia, 2013.

[3] E. Pogossian, V. Vahradyan and A. Grigoryan, "Experiments in consistency of chess expertise with decision making for etudes of Retie and Nodareishvili", *Transactions of IIAP NAS RA, Mathematical Problems of Computer Science*, (in Russian), vol. 28, pp. 94—113, 2007.

[4] M.M. Botvinnik, "Computers in chess: solving inexact search problems", *Springer Series in Symbolic Computation, with Appendixes*, Springer-Verlag, New York, 1984.

[5] E. Pogossian, A. Javadyan and E. Ivanyan, "Effective discovery of intrusion protection strategies", *The International Workshop on Agents and Data Mining, Lecture Notes in Computer Science*, St. Petersburg, Russia, vol. 3505, pp. 263-274, 2005.

[6] S. Grigoryan, "Structuring of Goals and Plans for Personalized Planning and Integrated Testing of Plans", *Mathematical Problems of Computer Science*, vol. 43, pp. 62-75, 2015.

[7] K. Khachatryan, S. Grigoryan and T. Baghdasaryan, "Experiments validating the Be-Have-Do meaning presentation model and matching algorithm for competing and combating problems", *International Conference in Computer Sciences and Information Technologies*, pp. 155-159, Yerevan, Armenia, 2013.

[8] K. Khachatryan and V. Vahradyan, "Graphical Language Interpreter Unified for SSRGT Problems and Relevant Complex Knowledge", *International Conference in Computer Sciences and Information Technologies*, Yerevan, Armenia, pp. 178-182, 2011.

[9] K. Khachatryan and S. Grigoryan, "Java programs for matching situations to the meanings of SSRGT games", *Proceedings of SEUA Annual conference*, pp. 135-141 Yerevan, Armenia, 2013.

[10] Grigoryan, Z. Naghashyan, M. Gurgiyan and E. Pogossian, "Strategy knowledge acquisition interface for computation anomalies dynamic analysis and correction", *International Workshop in Security*, pp. 76-86, 2010.