Implementation of Local Tournaments Methodology for Evaluation of Management Strategies

Emma Danielyan

Laboratory of Cognitive Algorithms and Models, IPIA,
National Academy of Science of Armenia,
Yerevan, Armenia
e-mail: emma_danielyan@yahoo.com

ABSTRACT

Problem of development, evaluation and ordering of competitive management strategy is described. A methodology for strategy generation and/or selection of the optimal one from predefined set of strategies is proposed. The methodology is based on the obtained in advance expert knowledge about simulated oligopoly market, generation of competitive strategies and continuous improvement of their performances relying on the dynamic testing results. Method of Local Tournaments is used as a central algorithm for the strategies' generation and ranking process. This method in conjunction with the strategy assessment package leads to creation or selection of the successful strategy. The suggested methodology for assessment of management strategies is consistent with PPIT-algorithm used for strategies assessment in chess and intrusion protection problems.

Keywords

Strategies assessment, simulation, local tournament, model

1. INTRODUCTION

In competitive environment it is natural to require acting in accordance with optimal strategy. Corresponding simulation models suppose that the solutions have to deliver recommendations on how to interpret the real world and how to act in it.

In the Management Optimal Strategy Provision (MOSP) problem a company is competing in oligopoly market according to some success criteria (max cumulative profit, max return on investment, etc.) and is going to make decisions in market situations that are consistent with the best strategy at least for defined periods of the competition [10].

To advance in the MOSP problem solution by simulating an adequate model it is required to create effective algorithms for strategy development and assessment that will be able to acquire regular management knowledge to increase the quality of the strategy.

In our model of oligopoly competition several companies compete for maximal profit. They form various strategies that describe qualitative changes of basic competition parameters - Price and Quality of goods produced by them [2].

The best strategy selection by Matrix of Tests is realized as a version of on-the-job performance assessment ideology based on the following three basic assumptions [9]:

1. the strategies are ordered on the absolute scale by their on-the-job integrated competition performances in all oligopoly competitions with all other competing strategies;
2. any two strategies may be ordered by the absolute scale using local computational resources;
3. there are adequate game models to simulate oligopoly competitions.

Many applications are based on the MOSP problem solution; particularly for constructing an advisor that will recommend decisions to a company in oligopoly competition or developing a scale for measuring a company's management strategy.

Most computer based methods to solve the MOSP problem are focused on constructing strategy plans. Advanced man-machine interactive tools already exist to help managers in planning and testing corresponding strategies by human teams' competitions in simulated environments. However, complete models including strategy plans generation, plans transformation into strategies followed by strategies static and dynamic testing in competitive environments are underdeveloped.

The Trading Agent Competition in Supply Chain Management game (TAC SCM) [3] is suggested as a model simulating the oligopoly competition.

2. OBJECTIVES

Management strategies development and testing framework with assessment and ranking methodology is proposed. The framework is a model of the knowledge-driven decision support system that is aimed to generate and recommend a competitive management strategy or a set of strategies that prove their intelligent behavior on the oligopoly competition.

The framework consists of simulation model for competitive environment, strategies assessment package, algorithms for generation of strategic plans and strategies. It is developed to create a scale consistent with on-the-job performance of management strategies and allowing to measure them in oligopoly competitions.

Based on the expert knowledge about simulated oligopoly market the framework allows generating competitive strategies and continuously improving their performance relying on the dynamic testing results. The method of Local Tournaments [8] is proposed as the main algorithm for the strategies' generation and ranking process. Initially this method was developed to assess optimal strategies in chess. Later it was proved applicable to the MOSP problem as well [10].

Creation of the framework for dynamic testing and assessment of management strategies provides an environment capable to answer essential management questions while developing an effective strategy:

- Where are we now?
- Where should we be?
- How do we get there?

3. INTRODUCTION TO THEORY OF TOURNAMENTS

3.1. Round-robin Tournament

Tournaments provide a model of the statistical technique called the method of pair comparisons. This method is applied when there is a number of objects, strategies in our case, to be assessed on the basis of some criterion. Based on the tournament results the objects are being compared and ranked.

A **round-robin tournament** T_n consists of n nodes $p_1 \ldots p_n$, such that each pair of distinct nodes p_i and p_j is joint by one

and only one of the oriented arcs $\overline{p_i p_j}$ or $\overline{p_j p_i}$. If the arc $\overline{p_i p_j}$ is in T_n, then we say that p_i dominates p_j and denote it as $p_i \succ p_j$ [6].
The score of p_i is the number s_i of nodes that p_i dominates. The score vector of T_n is the ordered n-tuple vector $(s_1 \ldots s_n)$.
If there are several objects that got the same score in the tournament it is suggested to consider the subtournament created by those objects and justify their scores based on it.
A tournament T_r is a **subtournament** of a tournament T_n if there exists an one-to-one mapping f between the nodes of T_r and a subset of the nodes of T_n such that, if $p \succ q$ in T_r then $f(p) \succ f(q)$ in T_n [6].

3.2. Uncertainty Zone

Uncertainty zone of the strategy p is a minimal range of score such that for arbitrary strategy g with score out of that range the strategy p always wins if *g's* score is less than that range and always loses if *g's* score is bigger [8].
Strategies **absolute tournament** is a round-robin tournament on the set of all initial situations $\Re$ such that in any situation $P \in \Re$ the pair of strategies meets twice exchanging the right of the first move. The set of all games in the absolute tournament that takes place between two strategies is called **match**.
Strategy f loses to strategy g ($f \prec g$ or $g \succ f$) if f's score in the strategies match is less than g's score. The result of such tournament is an absolute ordering O^* of the strategies set F_0 according to the strategy scores earned in the absolute tournament. The strategy won the first place on the tournament is called the **optimal strategy**.
Let $\lambda(f)$ is the strategy f's place on the tournament scale, $\lambda(g)$ is g's place and $\lambda(f,g) = \lambda(g) - \lambda(f)$ is the difference between these 2 strategies places on the absolute scale. If f strategy has a better place than g then $\lambda(f,g) > 0$. Two strategies are called **equivalent** if $\lambda(f) = \lambda(g)$.
Uncertainty zone for any arbitrary strategy f is a minimal scale range $[\lambda(f) - b_1(f), \lambda(f) + b_2(f)]$ where for any strategy g

1. $\lambda(g) > \lambda(f) + b_2(f) \Rightarrow f \succ g$
2. $\lambda(g) < \lambda(f) - b_1(f) \Rightarrow f \prec g$

Let's denote as b the value

$$\max_{f \in F_0} \max \{b_1(f), b_2(f)\}$$

Thus, b is the minimal length of the segment $[i, j]$ where any strategy gained the place i in the tournament always loses to the strategy gained the place j in the same tournament.

3.3. Method of Local Tournaments

The method of local tournament is based on the next sufficient criterion:
Theorem 1: if for arbitrary strategies f_1 and f_2, f_2 wins f_1 in match, the set of strategies F exists such that $|F| > 2b - 1$ and any $f \in F$ wins f_1 and loses f_2, then f_2 has a better (higher) place than f_1 in the absolute tournament [8].
Slight changes in the proof of this theorem bring to the next formulation:
Theorem 2: if for arbitrary strategies f_1 and f_2, there is a pair of strategies g_1 and g_2 such that $\lambda(g_1, g_2) > 2b - 1$ and both lose to f_2 and win f_1 then f_2 has a higher place than f_1 in the absolute tournament [5].
Based on the both described theorems the next 3 corollaries were proved.
Corollary 1: let there are strategies f_1 and f_2. If there is a set of strategies F, where $|F|>2b - 1$ and for any $f \in F$ it wins f_1 and loses to f_2, then $f_1 < f_2$.
Corollary 2: Theorem 2 is true in the case when strategy g_2 loses to strategy f_2 in the match and has a draw with f_1 ($g_2 \approx f_1$).
Corollary 3: let there are strategies h, g and f. It is known that strategy h wins g in the match and g wins strategy f. Let's suppose that $\lambda(g)$ is known. If there is a strategy g' such that $\lambda(g, g') > 2b - 1$ and strategy h wins strategy g' and strategy f loses to g, then $h > f$.

4. TOURNAMENT BASED STRATEGY ASSESSMENT

4.1. Maximal Sum Method

Integrative performance of management strategies in competitions can be evaluated by method of Maximal Sum (MS) that is similar to round-robin tournaments, where the best competitor is found by max sum of performances according to specified criterion K in all competitive oligopoly markets.
To advance in the MOSP problem solution assumptions A1 – A7 should be done [10]:
A1: On-the-job performance of management strategies by criterion K and maximal sum method induce an "ideal" ordering *O*(K,MS)* of all strategies.
A2: Each competitor is identified by a corresponding deterministic program.
A3: Competition in markets may be described by sets of situations, actions and strategies in discrete time periods.
A4: All competitors have the same sets of allowed market moves/actions.
A5: A competition of competitors $C_1, C_2, ..., C_m$ is determined by the sample of corresponding programs and by the initial situation.
A6: The quality of a competitor is evaluated by the set of strategies generated by corresponding program in all possible initial situations.
A7: The criterion K and the MS assessing method allow to order competitors in accordance with the *O*(K,MS)*.
Thus, to evaluate a competitor all its possible games against all possible samples of other competitors in all possible initial situations are considered. The number of competitors in samples depends on the assessment objectives. For oligopoly competitions, ideally, all possible combinations of competitors which are in the oligopoly should be considered and then they can be ordered in accordance with their performances.
The results of such tournaments can be presented by *mxn* matrix (**Matrix of grades**), where m and n are the numbers of analyzed competitors and all competitive market situations, correspondingly.
Based on the matrix of grades and MS method the competitors can be ordered and obtain an ordering consistent with the ideal ordering *O*(K,MS)*.
However, even for moderate values of the above parameters the evaluation of strategies by tournaments is computationally hard problem. Thus, to realize the idea in practice some additional constrains should be accepted.

4.2. Transitivity Constraint

Additional assumption should be done:
A8: Each competitor competes against all strategies including its own ones.
To get the ordering *O(K,M)* the matrix of grades can be created, where each row corresponds to a competitor and values in the row are determined by results of games that competitor had against all possible samples of competitors in oligopoly competition from all possible initial situations.
The analysis of matrix of grades is being reduced by using the following **Transitivity Constraint** (TC) [10]:
P_i and P_j are competitors and B_i and B_j are sets of samples of competitors losing to P_i and P_j, correspondingly. So,
P_i is stronger than P_j with respect to the ordering O(K,MS) if and only if B_i includes B_j.*
TC is based on the assumption that all competitors are essentially different in their skills.

To order P_i and P_j in the frame of the Transitivity Constraint it is enough to find a sample of competitors that lose to P_i and win P_j.

Note, that the Olympic and Swiss systems in sports are based on the similar criteria.

Unfortunately the TC assumption is too specific and strong for practical use.

4.3. Quasi-Transitivity Constraint

More general constraint than the TC was suggested in [10]. It can be used when competitors are close to each other by their strengths or even belong to the same class of equivalence in the *O*(K,MS)*.

If competitor *i* is stronger than competitor *j* in the *O*(K,MS)* let's denote by *B(i,j)* samples of competitive strategies that win *i* and lose to *j*. In other words, *B(i,j)* is the set of samples that in games with strategies *i* and *j* achieve results opposite to the fact that *i* is stronger than *j* in the *O*(K,MS)* ordering. Then *#B(i,j)* is the variance function for the fixed *i* and varying *j*, and it is the number of elements in *B(i,j)*.

If *#B(i,j)* for some competitors is not zero the Transitivity Constraint can be applied only with some errors. The relevance of the Transitivity Constraint is increasing if the distance between compared competitors *i* and *j* is increasing.

Quasi-Transitivity Constraint (QTC): Given strategies P_i and P_j, samples of strategies B_i and B_j losing to P_i and P_j, correspondingly, and the variance function *#B(i,j)*. It is possible to select constants *a* and *b* (small enough compared with the number of all strategies in the ordering *O*(K,MS)*) such that:

- if *j* belongs to the segment *[i+a, i-a]* then $\#B(i,j)<b+1$;
- if *j* does not belong to the segment *[i+a, i-a]* then $\#B(i,j)=0$ and P_i is stronger than P_j with respect to the ordering *O*(K,MS)* if and only if B_i includes B_j.

The QTC assumption allows formulate the following sufficient criterion for efficient assessment of management strategies that is consistent with Theorem 1 from 3.3.

Theorem: Assuming Quasi-Transitivity Constraint for ordering *O*(K,MS)*. Competitor *f* is stronger than competitor *g* (i.e. the location of *f* is better than *g* in the ordering *O*(K,MS)*) if *b* samples of competitors can be found such that *f* wins and *g* loses games against each of them [10].

This theorem formulates sufficient conditions to order any two strategies if we can indicate *b* competitive oligopoly markets such that the performance of one of the strategies is better than the performance of the other in all of these markets. Having the competitor *f*, the question of its strength relative to the competitor *g* is reduced to construction of a special tournament and estimation of parameter *b*. Even without an estimate of *b* with increasing the number of testing samples the likelihood of *f* to be stronger than *g* increases too.

5. OPTIMAL STRATEGY DEVELOPMENT FRAMEWORK

5.1. The TAC SCM Game

Supply Chain Management (SCM) is an approach of planning and coordinating the activities of organization across the supply chain, from purchasing raw material procurement to delivering finished goods to the end-customer.

The Trading Agent Competition Supply Chain Management game (TAC SCM) was designed to capture many of the challenges involved in supporting dynamic supply chain practices and provides a competitive benchmarking environment for developing and testing agent-based solutions to supply chain management. Autonomous software agents compete against each other as computer manufacturers in the oligopoly market: each agent must purchase components such as memory, hard drives and others from suppliers, manage a factory where computers are assembled, and negotiate with customers to sell computers [3]. The game has been designed jointly by a team of researchers from the e-Supply Chain Management Lab at Carnegie Mellon University, the University of Minnesota, and the Swedish Institute of Computer Science (SICS), with input from the research community [http://www.sics.se/tac/].

5.2. Software to Create Competitive Agent

TAC SCM Software consists of AgentWare – a sample of the trading agent, a Server for the TAC SCM game, and a simple game data toolkit for reading the game logs. The TAC SCM Server is an open-source server provided by the SICS for the Trading Agent Competition that should be used to run the developed trading agent on the server to test its strategy and qualify its behavior [http://www.sics.se/tac/].

TAC SCM Controlled Server is the tool described in the [12] that allows market conditions to be repeated across multiple games.

The Strategy Plans Builder is a tool developed for visual design and edition of strategy plans (SP) [1]. The Strategy Builder is provided by specific scripting language for definition, description and edition of SP. When the design or change of a SP is done the plan description is translated into the text representation according to specified structured format that is used by the tool.

Strategy Assessment Package is a tool for assessment and generation of the Optimal Strategy in the MOSP problem. It suggests the best strategy based on the matrix of grades induced by strategies testing [4]. Method of Local Tournaments [8] is used in the package along with well-known optimality criteria from voting theory such as Borda, Condorcet, Copeland, Simpson and multistage elimination tree [7].

Described above tools together with Local Tournaments Methodology are main components of the framework for optimal strategy development. Consequent improvement of strategies by injection of common knowledge and achievements from the management theory as well as individual experience from the experts and strategies dynamic testing by local tournament allow getting the optimal strategy.

5.3. Strategies Testing and Assessment

The method of Local Tournaments gives two approaches to the solution of the MOSP problem:

- the first is the ranking of given strategies according to their performance (optimality);
- the second one consists of creation of new strategies based on the given one and further comparison of the created and old strategies.

These are the part of suggested algorithm for development of optimal strategy for the trading agent.

In both cases the comparison of strategies is being done in accordance with the Quasi-Transitivity Constraint [10].

5.3.1 Strategies Competitive and Non-Competitive Testing

To find the best strategy for the company the strategies should pass through the "Non-Competitive" and then through the "Competitive" testing phases [2]. This approach we named as from Non-Competitive to Competitive Testing algorithm (NCCT).

Non-Competitive phase: after development of a strategy for the company (or for the agent) it should be tested in the oligopoly market. As a model of such market the TAC SCM game simulator is suggested [3]. In this phase only one agent with strategy under the test is run in the market. In case the agent's profit isn't positive to the end of the game the strategy is considered bad and needs to be improved.

Then the strategy is changed according to some criteria and the game is run again with the same agent in the same market but with updated strategy. To the end of the game the strategy is evaluated and the result is compared with the previous game's results.

These steps are repeated while the strategy that gains some money for the agent in the game will be created.

In other words, the strategy has to be perspective enough at least without any other competing strategy in the market.

Thus, at this phase of testing a row of strategies is generated. Then during the *competitive testing* phase all strategies that passed non-competitive testing will be ordered according to their on-the-job performance. In case the framework already contains some predefined set of strategies, which are already ordered and considered as an "ideal" ordering O*, the new generated strategies ordering will be isomorphically imbedded into that ideal ordering.

Competitive testing: the strategy should be compared with other strategies based on on-the-job performance simulation. For this phase at least 2 strategies should compete in the market. So, the game simulation is started with strategies which passed the non-competitive testing phase.

For more precise results it is recommended to run the same strategies against each other several times in the different market situations, but all strategies should be tested on the same set of market conditions [12]. Results of each simulation are recorded into the Matrix of Tests [2]. Having the test results matrix the Strategy Assessment package is used to order the strategies and identify the optimal one from the set induced by dynamic competitive testing.

To reduce the Matrix of Tests and avoid the comparison of one strategy with all already existing and ordered strategies the method of Local Tournaments is applied.

Sequential application of non-competitive and competitive testing along with the Strategy Assessment tool with build-in Local Tournaments methodology will generate and order different groups of strategies according to their on-the-job performance in the TAC SCM game.

5.3.2 Local Tournaments in NCCT - algorithm

Suggested market simulation model supports simultaneous run of 6 strategies in the market. Thus, Local Tournaments for our model are organized in accordance with the following basic rules:

1. all strategies passed non-competitive testing are divided by groups with 6 strategies in each;
2. each group of strategies competes in market on the same set of market situations;
3. Matrixes of Tests are created for every group of tested strategies;
4. strategies are ordered in accordance with their on-the-job performance in simulations and based on the tests' results.

After applying the mentioned 1-4 steps several sets of ordered strategies will be gotten. The following steps intend to place all strategies in one ordering *O**:

5. strategies from two sets of already ordered strategies should be compared by local tournaments and be included in one ordering *O**. It is obvious, that new ordering will be isomorphically imbedded to already existed orderings in both sets;

Local tournament is organized by the following rules:

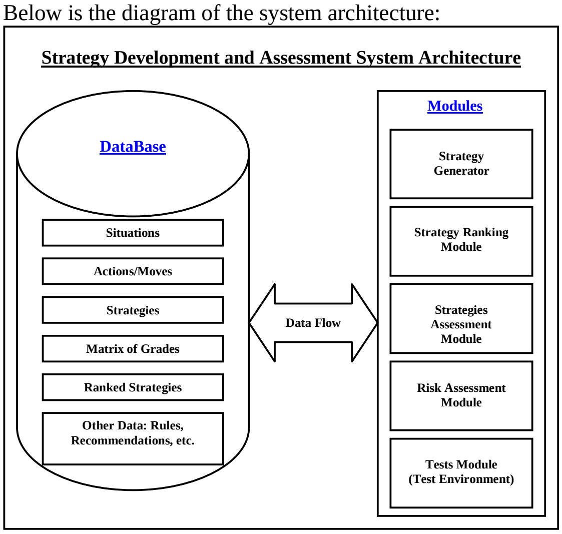
6. suppose we have as a result of previous ordering n $\{f_1, \ldots, f_n\}$ strategies in the first set and 6 strategies $\{g_1, \ldots, g_6\}$ in the second set. Note, that initially we have just 6 strategies in the first set of ordered strategies.
7. the first strategy g_1 from the second set is being compared with strategies from the first set;
   a. at first g_1 is compared with strategy f_p that has position $p=[n/2]$ in the ordering *O** in the first set;
   b. if $g_1 = f_p$, g_1 is placed just after f_p in *O**;
   c. if $g_1 > f_p$, it means that g_1 should be placed in the range $[1,p-1]$ and it is being compared with strategy that has position $p1=[p/2]$ in the first set;
   d. if $g_1 < f_p$, it means that g_1's place is in the range $[p+1,n]$ and it is being compared with f_{p2}, where $p2 = p+[n/4]$;
   e. depending on results in previous steps, strategy g_1 is comparing with strategies from the range $[1, p1]$, $[p1, p]$, $[p, p2]$ or $[p2, n]$ in accordance with steps *b-d* and the rule that just a half of the strategies from the defined range is considered for comparing. Process is being repeated while position of g_1 in the ordering *O** is found.
   f. when position of g_1 in *O** is found the strategy is added to the first set and comparison of g_2 is started. Only strategies that are placed on the right from g_1 in *O** is considered for comparison as in the second set $g_1 > g_2$. The same process described in steps *a-e* is used for g_2.
8. steps *a-f* are repeated for all strategies from the second set while these strategies are ordered in *O**.
9. after unifying and ordering all strategies from the first and second sets, the resulting set is considered as the set that will be compared with the next set of strategies containing other 6 ordered strategies.

Described process (steps 6-9) is repeated while all sets of generated and ordered strategies are unified in one ordered set.

5.3.3 Strategies Assessment Workflow

The core of the framework for optimal strategy development is the strategy development and assessment system that consists of database with necessary data to generate strategies and modules responsible for evaluation of market situations and allowable actions, strategy generation and evaluation.

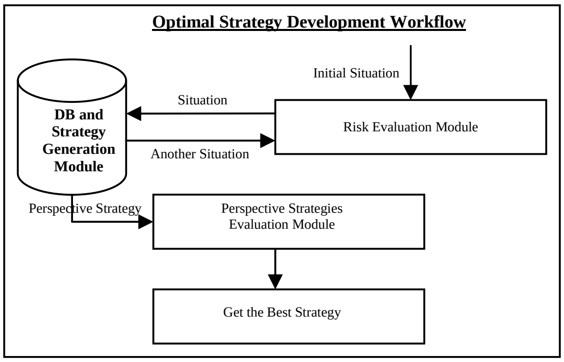
Below is the diagram of the system architecture:



Modules in the system are the software described in the 5.2. Data in the database can be input both manually by experts and automatically by the system according to its workflow algorithm.

The system provides user with options to generate, assess and rank the strategies.

Generalized presentation of the optimal strategy development workflow looks like:

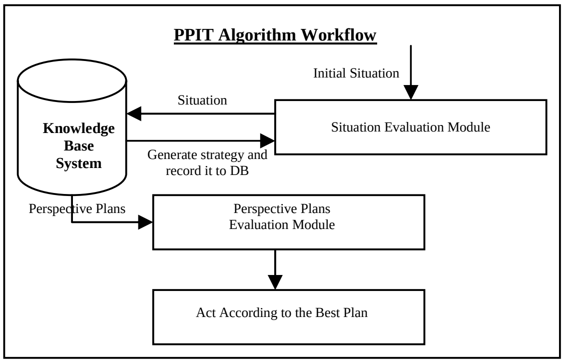


5.4. PPIT Algorithm

The MOSP problem along with network Intrusion Protection and Chess-like combinatorial problems belongs to *SSRGT* class where the *Space* of possible *Solutions* can be *Reproduced* by combinatorial *Game Trees*.

The PPIT (*Personalized Planning and Integrated Testing*) algorithms were suggested as a base for solving problems of SSRGT class [11]. They elaborate moves in target positions depending on the knowledge in the knowledge base system, which contains formal structures of attributes, goals, strategies, plans, etc.

By comparing the PPIT schematic representation on figure below with the above generalized presentation of strategy development algorithm we can state that the methodology used for the strategies development and assessment in the MOSP problem is consistent with PPIT-algorithm.



6. CONCLUSION

Methodology and framework for optimal strategy generation and assessment for the MOSP problem were suggested. Based on the expert knowledge about simulated oligopoly market the framework provides instruments for generating competitive strategies and continuously improving their performance by relying on the dynamic testing results in test environment.

The essential outputs of suggested methodology are

- generation of strategies with acceptable level of management decision making;
- continuous strategy improvement process;
- tool for strategy assessment based on their testing results;
- strategies ranking for particular competitive model in accordance with the absolute scale.

The method of Local Tournaments is described in terms of 3 theorems and 3 corollaries and is proposed for dynamic testing of management strategies.

Description of NCCT-algorithm is given as well. Strategies are generated and improved in the non-competitive testing phase while their assessment and ranking is done during the competitive testing. The results of competitive testing are used for organization of local tournaments and ordering the generated strategies.

The diagram with architecture of the framework system is given along with schematic presentation of workflow of the algorithm for the strategy assessment and ranking.

The PPIT algorithms developed for solution of SSRGT class problems and algorithm suggested for the strategy assessment and ranking were also presented schematically. Consistency of the strategy assessment approach with the PPIT-algorithm was shown in the paper.

7. ACKNOWLEDGEMENT

I would like to express my gratitude to Professor Edward Pogossian for supervision of the work and to Lev Dzhandzhulyan for his constructive comments to improve the paper.

REFERENCES

[1] T. Baghdasaryan, “Scripting Language and Design Tool for Trading Agents Strategy Plans”, *Proceedings of the Conference on Computer Science and Information Technologies CSIT2007*, Yerevan, Armenia, 2007.

[2] T. Baghdasaryan, E. Danielyan, E. Pogossian, “Testing Oligopoly Strategy Plans by Their On the Job Performance Simulation”, *Proceedings of the International Conference CSIT2005*, Yerevan, Armenia, 2005.

[3] J. Collins, R. Arunachalam, N. Sadeh, J. Eriksson, N. Finne, S. Janson, “The Supply Chain Management Game for the 2007 Trading Agent Competition”, *TAC SCM game specification for 2007*, http://www.sics.se/tac, 2007.

[4] E. Danielyan, “Software for Optimal Strategy Provision in Set Induced by Testing”, *Proceedings of the Conference on Computer Science and Information Technologies CSIT2005*, Yerevan, Armenia, 2005.

[5] L. Dzhandzhulyan, “Strategies Evaluation Criteria in Finite Positional Games and Models of Their Local Calculation”, *dissertation, Yerevan*, 1991 (in Russian).

[6] J. W. Moon, “Topics on Tournaments” *N.Y. etc,: Holt, Rinehart and Winston*, 1968.

[7] H. Moulin “Axioms of Cooperative Decision Making”. *ISBN-0-521-36055-2, Virginia Polytechnic & State University*, USA, 1988.

[8] E. Pogossian, “Adaptation of Combinatorial Algorithms”, *National Academy of Sciences*, Yerevan, 1983,293pp (in Russian).

[9] E. Pogossian, “Business Measurements by On-The-Job Competition Scale”, *International Conference “Management of Small Business: Problems, Teaching, Future”*, Sevastopol, Ukraine, 2004.

[10] E. Pogossian, “Models for Management Strategy Search and Assessment”, *Proceedings of the International Conference CSIT1999*, Yerevan, Armenia, 1999.

[11] E. Pogossian, V. Vahradyan, A. Grigoryan, "On Competing Agents Consistent with Expert Knowledge", *Lecture Notes in Computer Science, AIS-ADM-07: The Intern. Workshop on Autonomous Intelligent Systems - Agents and Data Mining*, June 5 -7, 2007, St. Petersburg

[12] E. Sodomka, J. Collins, M. Gini, “Efficient Statistical Methods for Evaluating Trading Agent Performance”, *Proceedings of AAAI-2007 conference,* 2007.