Appl. Comput. Math. **7** (2008), no.1, pp.89-106

TOWARDS REVERSE SCHEDULING WITH FINAL STATES OF SPORTS DISCIPLINES

VASIF V. NABIYEV †, HÜSEYIN PEHLIVAN †§

ABSTRACT. This paper addresses a tournament scoring problem that can lead to a score-based form of scheduling. Unlike classical combinatorial problems with some initial state and constraints, the problem is concerned with the construction of valid initial states according to some given final state and constraints and so it is named as *reverse scheduling*. Given a football tournament, this involves determining possible scores of all matches between teams, using the final state of the tournament table. For a feasible solution, the subject of multi-parameter partition is examined, the notion of *black&white* graphs is introduced, and a *rules-based* method is proposed. Also, using some particular table data, experimental results are presented along with the number of different scores and their computation time for one team.

Key words: scheduling, multi-parameter partition, black & white graphs.

1. Introduction

The evolution of powerful computers has made the solution of many large scale combinatorial problems possible. For example, the problem of animal squares (also called polyominoes) was solved after computers operated for ten months [16]. The enumeration of orthogonal Latin squares, could be solved only for order 10 with the assistance of more than 2,000 hours of computation on a supercomputer [12]. A concise discussion of enumeration problems can be found in [13]. For certain problems that cannot be practically solved with classical solution methods, it is possible to perform combinatorial computations using a knowledge base. For example, the knowledge-base approach adopted for problem analysis in the paper always discards many unnecessary cases, which is the most important difference from heuristic ones such as A* [8]. In our approach the initial state of the problem is repaired according to the results of its solution. To illustrate, we will discuss a solution algorithm of the *tournament scoring problem*, which belongs to a different class of problems in terms of problem analysis and requires to be computed combinatorially.

In recent years sports scheduling has also become one of the major areas of computer applications. Many different models and algorithms have been developed for automated scheduling of various sports disciplines, such as soccer [2, 5], basketball [6, 9], icehockey [3, 7] and cricket [1, 18]. Typical examples of the models include graphs, round-robin tournament, travel distances and playoff/first place elimination. The algorithms are usually based on approaches such as simulated annealing, constraint logic programming, linear programming, and graph coloring heuristics. However, the main concern of scheduling algorithms is on soccer, possibly due to its popularity, where the scheduling process deals with matches played in both league and tournament environments.

Each sports discipline has its own set of rules and regulations. Moreover, there might be many variations of rules in a particular sports discipline because every nation has their own inner-league requirements and preferences. Generally, in sports competitions, teams play against each other according to a certain scheme which may be country-dependent. The most popular scheme in team sports like soccer and basketball is round robin, where every team plays against every other team a fixed number of times during the competition. Conventionally, if this number equals 2,

---

†Department of Computer Engineering, Karadeniz Technical University, 61080, Trabzon, Turkey,
E-mail: vasif@ktu.edu.tr, pehlivan@ktu.edu.tr

§*Manuscript received 26 September, 2007.*

the competition corresponds to a sports league in which a team plays home and away matches, while a one-match scheme corresponds to a sports tournament.

The current literature includes many studies on round-robin tournament scheduling [4, 10, 14, 17]. However, football competitions can also be conducted in other types of tournaments, such as elimination, consolation and challenges. In this paper, we suppose that teams play each other in a round-robin tournament. Unlike traditional scheduling problems, the process of scheduling football tournaments is reversed, trying to determine possible scores of matches played by each team using their final positions. This has not been previously investigated in the literature. The initial data of the *reverse scheduling* is taken from the final table of a football tournament, which contains the following values for each team:

- the number of wins (W)
- the number of draws (D)
- the number of losses (L)
- the number of goals for (F)
- the number of goals against (A)

In fact, by "reverse scheduling" we mean that the tournament is rescheduled together with possible scores of all played matches such that the scores would meet the final values W, D, L, F, A for each team. From this point of view, there is a noticable difference between the usual scheduling and reverse scheduling. The former is inherently carried out in the domain of several parameters such as time and venue. The latter, additionally, involves considering the possible scores of each match. In this way the reverse scheduling is assumed to consist of three stages:

- computing possible match scores for every team in accordance with W, D, L, F, A,
- determining those scores that establish match-based relations among teams,
- scheduling the tournament for each valid set of scores.

These stages make the reverse scheduling more difficult than the usual problem, generating a large solution space. To decrease the solution space, the scores of some matches can be included into the initial data set of the problem. Our concern is restricted to determining the possible results and scores of all matches among teams attending a football tournament, depending on the final tournament table. The inputs of the problem are the number of wins, ties and loses as well as the total number of goals scored and conceded by each team. First let's consider a simple instance associated with the problem.

2. Problem Description

International football tournaments like the World Cup often begin with group matches in the first round before entering the quarter finals for a straight knockout. There are usually four groups with four teams each. Each team plays each of the other teams in the same group once. In order to describe the problem, we will use the following final table, which might belong to such a group in the tournament, with the aim of finding the scores of all the matches played in the tournament (see Table 1). Each team has played one match against the other three teams, totaling six matches. The only constraint imposed on the problem is that team B has won against team A.

Let's try to manually find the solution of the problem by going directly through the table. The statistics associated with team D is the best starting point of the manual solution, because the values $W = 0$ and $F = 0$ make it easy. Since beating team A, team B must have drawn with teams C and D. Similarly, since team D scored no goal, the matches of teams B and D must have ended in a draw of $0 - 0$. Now, suppose that the score of the draw between teams B

Table 1. Final table of a group with four teams in a football tournament

Team	P	W	D	L	F	A	GD	PTS
A	3	2	0	1	5	3	2	6
B	3	1	2	0	4	3	1	5
C	3	1	1	1	5	3	2	4
D	3	0	1	2	0	5	-5	1

and C had been $X-X$. In this case, as team B scored four goals and conceded three goals, the match of teams B to A must have resulted in a score of $(4-X)-(3-X)$.

Since team A lost against team B, it must have won its other two matches. Team A has conceded $(4-X)$ goals from team B and no goals from team D. Thus, if $(4-X)$ goals of team B are subtracted from a total of three goals scored by team A, it becomes obvious that team A has conceded $3-(4-X)=(X-1)$ goals from team C. Now that team C has a total of five goals, of which $(X-1)$ have been scored against team A and X have been scored against team B, team C must have scored $5-(X-1)-X=(6-2X)$ goals against team D. Additionally, since team D, which has a total of five goals, conceded no goals against team B, it must have allowed $5-(6-2X)=(2X-1)$ goals from team A. Finally, from a total of five goals of team A, $5-(3-X)-(2X-1)=(3-X)$ goals have been scored against team C.

Table 2. Possible scores of the group matches

Match	Relative score	Actual score
B-D	0-0	0-0
B-C	X-X	1-1
B-A (B wins)	(4-X)-(3-X)	3-2
A-C (A wins)	(3-X)-(X-1)	2-0
A-D (A wins)	(2X-1)-0	1-0
C-D (C wins)	(6-2X)-0	4-0

Table 3. Results of the group matches (1 means a win, 0 to a draw, and 2 to a loss)

Team	A	B	C	D
A	–	2	1	1
B	1	–	0	0
C	2	0	–	1
D	2	0	2	–

From the fourth and fifth lines of Table 2, it can be seen that there are two equations of inequalities, $(3-X)>(X-1)$ and $(2X-1)>0$, respectively, because team A has won against teams C and D. The only solution to these inequalities is $X=1$. Using this value, the scores of all the six matches can be found, which is given in Table 2. The results of the matches are shown in Table 11, representing wins, draws and losses with the numbers 1, 0 and 2, respectively.

As a result, the solution of the problem usually involves solving two subproblems, exactly as above. That is, the first one is to determine wins and draws of teams according to the number of matches played between two teams (Note that two teams play each other in a group of the tournament only once, while they play home and away matches in a league). The second subproblem is to determine possible scores of matches according to wins and draws of teams. The number of teams involved in the main problem is obviously the most important factor in the solution of the subproblems. The tournament table does show only the number of wins and draws of a team, but not which other teams it wins against and draws with. In a group with four teams, it is always easy to find wins and draws of each team. However, this gets more difficult as the number of teams increases and their home and away matches are of interest, resulting in the inapplicability of manual solution methods.

3. A General Approach to the Problem

In the problem, if each team is represented by a node, we get a regular graph at the end of group matches. The total number of edges in undirected regular graphs is given by

$$E = \frac{1}{2}\sum_{i=1}^{n} S_i.$$

The issue of representing this regular graph with isomorphic matched pairs of its vertices is known as the scheduling problem. An instance of such graphs for eight teams in the scheduling problem is shown along with a possible schedule in Fig.1, which consists of seven levels and six different sets [11]. Each level of the schedule tree gives a proper matching of teams, while each path from root to leaf gives one tournament schedule of seven rounds (that is, each round is represented with a single node). The matches played in each round is represented by the leftmost path of the tree is given in Table4.

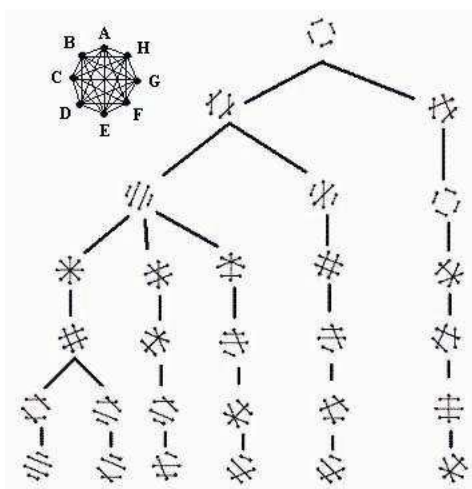

Figure 1. The regular graph and six possible tournament schedules for eight teams

Table 4. A possible schedule of all seven rounds in a tournament with eight teams

Week	Match 1	Match 2	Match 3	Match 4
1	A-B	C-D	E-F	G-H
2	A-C	B-D	E-G	F-H
3	A-D	B-C	E-H	F-G
4	A-E	B-F	C-G	D-H
5	A-F	B-E	C-H	D-G
6	A-G	B-H	C-E	D-F
7	A-H	B-G	C-F	D-E

TABLE 5. The number of home and away matches, and their possible results according to the number of teams

Number of teams (n)	Number of matches (home and away) (m)	Possible results
2	2	$3^2 = 9$
3	6	$3^6 = 729$
4	12	$3^{12} = 531441$
5	20	$3^{20} = 3486784401$
6	30	$3^{30} = 205891132094649$
7	42	$3^{42} = 109418989131512359209$
8	56	$3^{56} = 523347633027360537213511521$
9	72	$3^{72} = 22528399544939174411840147874772600$
10	90	$3^{90} = 8,727963568087712425891397479476 7 * 10^{42}$

In general, given home and away matches of each team in a league with n teams, there are $(m = n * (n-1)$ matches. Since a single match can have three different in-game results such as wins, draws and loses, the number of all possible results of the matches are given as 3^m. For example, in the Turkish football league with 18 teams, there are a total of 306 matches and thus a total of $3^{306} \simeq 10^{46}$ results (see Table 5).

Even if every team played one match against the other teams, as in tournament environments, the total number of played matches would be $(n * (n-1)/2$ and so the problem would continue to keep its exponential character. For example, in a one-match competition with three teams, a total of three matches can yield a set of 27 different results. For three teams named A, B and C, this set is shown in Table 6 representing a win, draw and loss of a team with 1, 0 and 2, respectively.

TABLE 6. All possible results of the matches in a one-match competition with three teams

No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	...	25	26	27
A-B	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	...	0	0	0
A-C	1	1	1	2	2	2	0	0	0	1	1	1	2	2	2	0	0	0	...	0	0	0
B-C	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	...	1	2	0

Now, let's focus on the set of match results associated with a single team. For a team playing n matches, the number of possible results can be computed by the recurrence formula

$$f(0) = 1,$$
$$f(n) = f(n-1) + n + 1.$$

For example, the result of one match ($n = 1$) corresponds to one of the WDL values in the set $\{001, 010, 100\}$, the number of which is determined by $f(1)$. The WDL value of 001 means that the team has a loss, but not a win or draw. Similarly, the WDL value of 010 means a draw and the WDL value of 100 is a win. A team playing two matches ($n = 2$) can have a set of six WDL values, namely $\{200, 110, 101, 020, 011, 002\}$. This can be obtained by interpreting the values on the columns of Table 6. The interpretation related to team A is shown in Table 7, which can also be done for teams B and C. Note that the repeated WDL values are not shown as bold in Table 7, in order to keep the others unique.

For $n = 3$, ten possible results are

300 210 111 201 120 030 021 012 102 003

TABLE 7. The WDL values derived from Table 6 for team A

No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	...	25	26	27
A-B	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	...	2	2	2
A-C	1	1	1	0	0	0	2	2	2	1	1	1	0	0	0	2	2	2	...	2	2	2
WDL	**200**			**110**			**101**			110			**020**			**011**			...	**002**		

For the values $n = 0, 1, 2, ..., k$, the function $f(n)$ generates the triangle numbers, such as $1, 3, 6, 10, ..., k * (k+1)/2$. Its closed form can be given by

$$f(n) = T_n = \frac{(n+1)(n+2)}{2} = \begin{pmatrix} n+2 \\ 2 \end{pmatrix}$$

.
The complexity of this function is obviously $O(n^2)$. Since, in a tournament with n teams, every team plays $(n-1)$ matches, the resulting complexity would be $O(n^{2(n-1)}) \simeq O(n^{2n}) \simeq O(n^n)$. This is known as geo-combinatorial complexity.

As seen above, the number of goals scored and conceded determines the score of a match. The same result can be reached in many combinations of the goals scored by the rival teams. That is, each of the scores $1-0$, $2-0$ and $2-1$ means either a win or loss in terms of the match result. Accordingly, the tournament scoring problem imposes a multi-parameter partition of the goals scored and allowed.

4. Multi-Number Partition

Many combinatorial problems are closely related to the classical problem of partitioning numbers. A number partition problem is basically described by the following question: What is the number of ways of writing the integer n as a sum of positive integers? This can also be formulated as follows:

Let $b_1, b_2, ..., b_k (k > 0)$ denote positive numbers. The number of the different partitions of a integer n is searched such that $n = b_1 + b_2 + ... + b_k$.

The order of the terms in the above formula is not important. For example, each of the additions $(1 + 2 + 2)$, $(2 + 1 + 2)$ and $(2 + 2 + 1)$ is the same. These terms are generally selected from a consecutive sequence $\{p_1, p_2, ..., p_k\}$, where $p_1 \leq p_2 \leq ... \leq p_k$. The set of each consecutive numbers $\{p_1, p_2, ..., p_k\}$ is considered as a sum of exactly k terms. Let $P(n)$ denote all the partitions of the number n, and $P(n, k)$ denote the k partitions of the number n. The former is given by

$$P(n) = \sum_{k=1}^{n} P(n, k), \qquad n > 0,$$

where $P(0) = P(0, 0) = 1$. The latter is given by a recurrence relation

$$P(n, k) = P(n-1, k-1) + P(n-k, k).$$

The common feature of classical number partition problems in combinatorics is that they involve a one-dimensional partition of numbers with different conditions. The tournament scoring problem appears as a two-dimensional partition one of the numbers in the WDL value (a triple) and the FA value (a pair), because all the match scores must totaly meet both the numbers W, D, L, and F, A.

To see this, let's examine the scoring problem further in terms of the partition theory of numbers. At the end of n matches a team plays, suppose that the scores of the matches are included in the three separate sets

$$\begin{aligned} S_W &= \{(h_i, a_i) \mid i \in \{1, 2, 3...n\}, h_i > a_i\}, \\ S_D &= \{(h_i, a_i) \mid i \in \{1, 2, 3...n\}, h_i = a_i\}, \\ S_L &= \{(h_i, a_i) \mid i \in \{1, 2, 3...n\}, h_i < a_i\}, \end{aligned}$$

where h_i and a_i denote the number of the goals scored by the home and away teams in the ith match. The number of the pairs in the sets S_W, S_D and S_L equals to the values W, D, and L in the tournament table, respectively. Thus, for a team T, we can represent the scores of all the matches with the set $\{(h_1, a_1), (h_2, a_2), ..., (h_n, a_n)\}$. That is,

$$\begin{aligned} S(T) &= S_W \vee S_D \vee S_L, \\ S(T) &= \{(h_i, a_i) \mid i \in \{1, 2, 3...n\}\}. \end{aligned}$$

Now, suppose that the host team has a total of F-goals scored and A-goals conceded. The set S must satisfy the following equations

$$\begin{aligned} F &= h_1 + h_1 + ... + h_n, \\ A &= a_1 + a_1 + ... + a_n. \\ n &= W + D + L. \end{aligned}$$

This proves that the tournament scoring problem is a two-dimensional number partition one, where both the numbers in the WDL and FA values are partitioned into the sets S_W, S_D and S_L, computing the pairs (h_k, a_k), $k = 1, 2, 3, ..., n$. So the scoring problem is *multi-partition*.

Using the above definitions, we can express the scoring problem in another way; it consists of finding a relation based on scores between all teams. Given the tournament table in Section 2, the teams would have the following valid set of scores

$$\begin{aligned} S(A) &= \{(a_1, b_1), (a_2, c_1), (a_3, d_1)\}, \\ S(B) &= \{(b_1, a_1), (b_2, c_2), (b_3, d_2)\}, \\ S(C) &= \{(c_1, a_2), (c_2, b_2), (c_3, d_3)\}, \\ S(D) &= \{(d_1, a_3), (d_2, b_3), (d_3, c_3)\}. \end{aligned}$$

These scores belong to the matches in a football tournament participated by teams A, B, C, and D, because there are score relations between all the teams. For example, the relation between teams A and C is established by the scores of $a_2 - c_1$ and $c_1 - a_2$, respectively. This has a remarkable issue; namely, the set of scores for team D can be derived from the valid ones for other teams.

5. Problem Verification and Result Calculation

The problem has some practical implications for sports organizations. First the source data (e.g., the results of rounds) might not be available for some competitions that were held long ago. Our approach has the potential to calculate the source data of such a competition from its final table. Secondly a sports league generally demotes a team and promote another at the end of the season, mostly based on on-field performance. Therefore, it is quite important to find out in advance what results of rounds ensure a sports team to gain a dominant position that, for example, promotes it to a higher league. This is achievable based on some statistical data. For a certain sports league, a statistical table can be obtained by examining the final tables of the league over past years. In an intermediate round of the league, a new table is generated by calculating the difference between the corresponding entries of the statistical and intermediate tables. Then, using the data in this new table, our approach would allow one to predict both the results and the scores of the remaining matches.

Another example to the practical applications is sports scheduling which has an economic aspect. Professional sports are big business and the revenue of a sports league may be affected by the quality of the schedule since a substantial part of the revenue often comes from TV networks. The TV networks buy the rights to broadcast the games but in return they want the most attractive games to be scheduled at certain dates [15]. In this way, the problem of reverse scheduling can be associated with a score-based arrangement of football competitions. We can schedule the competitions in a football league according to the outcomes of the previous year. The attempt of maintaining a balance in the competitive strength of the teams would lead to a schedule of more attractive competitions. This paper focuses on the first two of three solution stages proposed for the problem.

The data residing in the final table of a football league is the initial one of the tournament scoring problem. A final table with correct data offers one or more possible solutions to the problem. Therefore, before searching for a computer-based solution, the table data must be verified for all teams of the league to ensure that it meets the following prerequisites.

- Each team in a league with n teams must play $2 * (n-1)$ matches (home and away ones with the other teams).
- The total number of goals scored by all the teams must equal to the total number of ones conceded.
- The total number of draws in the table must be even.
- The total number of wins in the table must be equal to the total number of losses.
- Every team must score a number of goals at least equal to the number of won matches.
- Every team must concede a number of goals at least equal to the number of lost matches.

In general, some kind of graphs can be used to determine the match results in the problem. For instance, the most suitable graph for four teams ($n = 4$) looks like 9-mixed tournament one as seen in Fig.2. Each node in the graph, whose degree is computed by $3 * (n-1)$, contains edges representing wins, draws, and losses for a team. A win or loss is shown by a directed edge (that is, the outgoing or ingoing edge, respectively) and a draw is shown by a undirected edge. The solution for the problem resides on a collection of 3-mixed tournament graphs. The total number of 3-mixed graphs is equal to the number of possible results. Given four teams, this number will be 27, which is enumerated in Section 3.

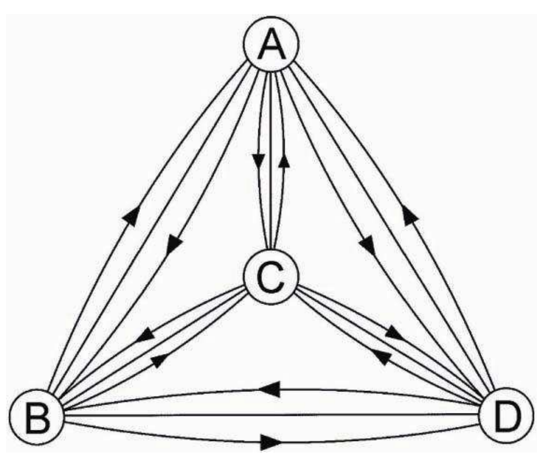

FIGURE 2. A 9-mixed tournament graph

In Section 2, we have manually found that a tournament scoring problem with the given final table has only one solution. But this is not possible all the time. For example, if team D had one win or scored at least one goal, the number of possible solutions would increase, making the problem more complicated. However, with the use of computers, we can determine if a particular match results in a draw or, if the result is not a draw, who wins the match. The corresponding 3-mixed graph is given in Fig.3.

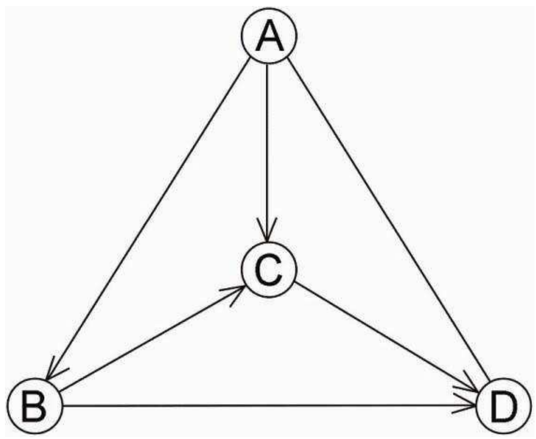

FIGURE 3. The 3-mixed tournament graph

The calculation of the results requires the partitioning of WDL values among the related matches. We call each different partition a *result case* (or shortly RC) and simply represent it by a combination of possible match results. In other words, for a given WDL value, each different combination of W 1s, D 0s and L 2s corresponds to a RC. The number of various partitions of the WDL value (i.e. the number of RCs) for a team T playing n matches is given by

$$P_{WDL}(T) = \frac{n!}{w!\,d!\,l!} = \frac{(w+d+l)!}{w!\,d!\,l!}.$$

For example, after playing six matches, if a team has the WDL value of 312 (i.e., 3 wins, 1 draw, and 2 losses), it could gain this value from $6!/3!1!2! = 720/12 = 60$ different RCs. For simplicity, let us give the RCs for team A with the WDL value of 201. It contains $P_{WDL}(A) = 3!/(2!0!1!) = 3$ RCs, namely 112, 121 and 211. From these cases, for example, the RC of 121 means that the first and last matches are won, and the second match is lost. All the RCs for the teams given in Table 1 are shown in Table 8.

As one can see, RCs are considerably different from WDL values, although they seem to be the same in terms of values. A RC reflects the result of each played match, while a WDL value shows the number of wins, draws and losses. So the size of a RC (i.e. the number of its components) equals to the number of the played matches. The components of each RC appear in the same order that the rival teams of their corresponding matches appear in Table 8.

A solution of match results comes with the selection of an appropriate RC of each team. All the result solutions can be obtained by using a component-based comparison of the RCs of the teams. To show this, let us assume that teams A and B played their first match against each other and team A lost the match. The first component of the RCs for teams A and B has to be 2 and 1, respectively. This means that a component of 1 in RCs of a team changes to a

TABLE 8. Possible RCs for the teams given in Table 1

Team	WDL	RCs
A	201	112, 121, 211
B	120	100, 010, 001
C	111	102, 120, 012, 021, 201, 210
D	012	022, 202, 220

component of 2 in those of other teams, and vice versa. A component of 0 stays unchanged through the cases. Given the RCs in Table 1, the first component of the RCs of team A cannot be 1 because any RC of team B has no component of 2 (i.e. it has no loss).

A blind search algorithm operating on the RCs listed in Table 1 would perform $P_{WDL}(A) * P_{WDL}(B) * P_{WDL}(C) * P_{WDL}(D) = 3*3*6*3 = 162$ evaluations. However, with the consideration of the component-based relations between the RCs, the number of the evaluations would be much less. In Table 8, the RCs can first be selected for the matches of team A because it is one of the teams for which P_{WDL} is minimum, which would reduce the number of the RCs to analyze. For example, when the result of match $A - B$ is selected as 2, there remains only $P_{WDL}(B) = 2!/(0!2!0!) = 1$ possible RC for the results of the remaining two matches ($B - C$ and $B - D$) of team B, that is, they must be all draws. Note that the match results for the last team can be obtained from the other teams. In this way, Table 9 illustrates how to calculate the possible results, using the RCs presented in Table 8.

TABLE 9. A RC evaluation algorithm for results of matches among the teams

Step	RC of A	RC of B	RC of C	RC of D	Reason
1	$\underline{1}12$	–	–	–	Unmatched
2	$\underline{1}21$	–	–	–	Unmatched
3	$\underline{2}11$	$\underline{1}00$	–	–	Matched
4	$2\underline{1}1$	100	$\underline{2}01$	–	Matched
5	211	$1\underline{0}0$	$2\underline{0}1$	–	Matched
6	$21\underline{1}$	100	201	$\underline{2}02$	Matched
7	211	$10\underline{0}$	201	$2\underline{0}2$	Matched
8	211	100	$20\underline{1}$	$20\underline{2}$	Matched

In the RC evaluations, the selection of each RC for a team affects the selection of RCs of the other teams, based on the existence of the related components in them. For example, the selection of the RC of $\underline{2}11$ for team A allows the selection of only the RC of $\underline{1}00$ for team B. It also allows two alternative selections of the RCs (i.e. $\underline{2}01$ or $\underline{2}10$) for team C, but the RC of $1\underline{0}0$ for team B makes the selection of only the RC of $2\underline{0}1$ possible. The solution is reached by selecting one RC for each team so that all the components in the selected RCs can be matched together in pairs. Note that the matched components in the RCs of two teams are underlined in each step specified in Table 9.

6. Score Calculation

The RC evaluation algorithm can only be used to disclose the match results. One more step is required to compute the match scores. To keep the size of the state space small, the results are used in the computation process of the scores. On generating this state space, the following conditions must be considered.

(1) A team has to score at least one goal for each win. So the maximum number of goals it can score in one match is $F - (W - 1)$. Similarly, the maximum number of goals it can concede in one match is the minimum of the values $F - W$ and $A - L$.

(2) A team has to concede at least one goal for each loss. So the maximum number of goals it can concede in one match is $A-(L-1)$. Similarly, the maximum number of goals it can score in one match is the minimum of the values $F-W$ and $A-L$.
(3) In the case of a draw, the maximum number of goals scored or conceded in one match is the minimum of the values $F-W$ and $A-L$.
(4) When the goals scored and conceded are subtracted from the table, the resulting table has to contain as many goals as it satisfies the winning and losing conditions.

Using these conditions, it is possible to generate the state space containing only the possible results, and to determine the valid scores of the matches through it, depending on the table data of each team. Besides, for a computerized solution of the problem, the entire state space can also be used to compute all possible scores of wins, draws, and losses for each team. For example, suppose that possible scores of wins, draws, and loses are to be computed in the specified order. In this case, while computing the drawing scores after doing the winning scores, the minimum value in $\{F-W, A-L, F-F_W, A-A_W\}$ can be taken as the maximum number of goals scored or conceded in one match, where F_W and A_W are, respectively, the total numbers of goals scored and conceded in the won matches. Finally, while computing the losing scores, the minimum of the values in $\{F-W, A-L, F-F_{WD}\}$ can be taken as the maximum number of goals scored in one match, where F_{WD} is the total number of goals scored in the won and drawn matches. Similarly, new short-cut values can be found for different computation orders of the possible winning, losing, and drawing scores, which is not discussed further in this paper.

The above conditions restrict the number of goals scored and conceded in a single match. Therefore, we can consider the values W, L, F, and A as constraints and say that, for example, the highest winning score will be able to be $(F-W+1)-(A-L)$ for any team with at least one win (i.e. $W>0$). For team B specified in Table 1, the highest winning score can be determined to be $(4-1+1)-(3-0)=4-3$. This does not mean that team B can have any winning score in the range of $1-0$ to $4-3$. For example, the score of $4-2$ or $4-1$ is not a valid winning one for team B, because otherwise one of the remaining scores would have been $0-1$ or $0-2$ but the team has no loss. The invalidity of such scores must be determined dynamically while F and A are being partitioned according to the WDL value. During the calculation of winning scores, the domain of both the components of the score can be discrete or continuous. Similar cases can occur for drawing and losing scores, too. We use some form of constraint programming in the calculation of match scores.

Using the data contained in Table 1, let us demonstrate the steps of how to calculate match scores of each team and generate a corresponding regular black&white graph. The fact that the match $B-D$ ends with a draw and team D has scored no goal means that it can have only a score of $0:0$. After imposing the above constraints on the state space, the method to be applied to search in this space is the same as the approach with which the winner of a match is determined. Although there are three cases to determine the winner of each match, the number of cases associated with the score of a match is unstable. For example, Table 1 shows that team A has played 3 games, gaining 2 wins and 1 loss, and scored 5 goals and conceded 3 goals. Representing the match scores for this team by the set $\{(s_1-c_1),(s_2-c_2),(s_3-c_3)\}$, all possible scores can be computed by means of the team-specific relations

$$s_1+s_2+s_3=F=5\ ,\qquad c_1+c_2+c_3=A=3.$$

Assuming that the first two elements of the set correspond to winning scores and the last element to a losing score, we can establish another relations

$$s_1-c_1\leq(F-W+1)-(min(F-W,A-L))=(5-2+1)-(min(5-2,3-1))=4-2,$$
$$s_2-c_2\leq(F-W+1)-(min(F-W,A-L))=(5-2+1)-(min(5-2,3-1))=4-2,$$

$$s_3 - c_3 \leq (min(F - W, A - L)) - (A - L + 1) = (min(5 - 2, 3 - 1)) - (3 - 1 + 1) = 2 - 3.$$

The solutions of these relations are listed in Table 10.

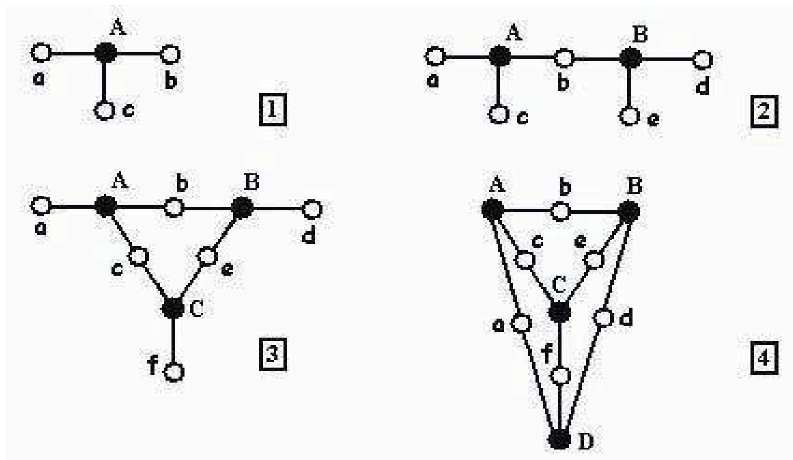

FIGURE 4. 3-mixed tournament graph generated with 4-team combinations

A solution strategy to cope with the problem is given for four teams named A, B, C, and D in Fig.4, in which the numbers indicate the order of the 4-team combinations. It can be theoretically shown by a graph with degree 3 because each of n teams plays $n-1$ games (i.e., one game with each of the other $n-1$ teams). Let the symbols a, b, and c denote the scores of matches between team A and the other ones (e.g., Table 11 says that these three possible scores would be $\{a, b, c\} = \{1-0, 2-0, 2-3\}$). After the first selection, a new rule-based one is made from the set of the scores of team B such that $A \cap B = \{a\} \vee \{b\} \vee \{c\}$. If $B = \{b, d, e\}$, the search for the selection of the next team, C, will be made among the unmatched elements of team A or B, which reside in the set $M_{AB} = (A \cup B) \backslash (A \cap B) = \{a, c, d, e\}$, such that $C \cap M_{AB} = \{a, d\} \vee \{a, e\} \vee \{c, d\} \vee \{c, e\}$. If $C = \{c, e, f\}$, since $M_{ABC} = (C \cup M_{AB}) \backslash (C \cap M_{AB}) = \{c, e\}$, there is one selection left for the last team, namely $D = \{a, d, f\}$. If this data is encountered in the set of scores of team D (that is, if $M_{ABCD} = (D \cup M_{ABC}) \backslash (D \cap M_{ABC}) = \emptyset$), the process stops and the next solution is similarly searched. Otherwise, going back, the other alternatives are examined in a rule-bound way.

As seen above, the solution starts from a uniformly structured node with degree $n-1$, and ends when a regular graph is found by establishing bilateral relations according to the common nodes. The arising graph is called *black&white* (*b&w*) graph in the problem. Black nodes represent teams, white represents the number of games. A regular graph can be produced by combining common white nodes. We call *vertice chaining* the combination of unique structured subgraphs to generate this regular black&white graphs. The following example demonstrates the steps of how to do this, using the data contained in Table 1.

According to Table 1, the fact that the match $B-C$ ends with a draw and team C has scored no goal means that it can have only a score of $0-0$. After imposing the above restrictions on the state space, the method to be applied to search in this space is the same as the approach with which the winner of a match is determined. Although there are three cases to determine the winner of each match , the number of cases associated with the score of a match is unstable. For example, Table 1 shows that team A has played 3 games, gaining 2 wins and 1 draw, and scored 5 goals and conceded 3 goals. The set of possible scores for this team are computed as given in Table 10.

TABLE 10. Possible scores of matches for team A

No.	M1	M2	M3	No.	M1	M2	M3
1	**1-0**	**2-0**	**2-3**	8	2-0	2-1	1-2
2	1-0	3-0	1-3	9	2-0	3-0	0-3
3	1-0	3-1	1-2	10	2-0	3-1	0-2
4	1-0	4-0	0-3	11	2-0	3-2	0-1
5	1-0	4-1	0-2	12	2-1	3-0	0-2
6	1-0	4-2	0-1	13	2-1	3-1	0-1
7	2-0	2-0	1-3				

TABLE 11. Selection of the scores for team A

Match	Result	Score	Action
A-B	2	2-3	Continue
A-C	1	2-0	Continue
A-D	1	1-0	Continue
...	...	...	...

The resulting set actually contains a total of 25 scores, but eliminating symmetrical cases (i.e., selecting either $\{(a,b),(c,d),(e,f)\}$ or $\{(c,d),(a,b),(e,f)\}$) decreases this number to 13. Without sorting among all the computed scores, only the use of the rules causes a relatively blind search which involves examining each of these 13 scores as well as the ones of the other teams. For instance, taking a score of $2-3$ for the match $A-B$ from the state space, the searching process continues as in Table11.

It can also generate more than one appropriate score, depending on the table data of the team. Giving the scores of some matches to the computer application, the size of the state space can be decreased.

TABLE 12. Possible scores of matches for teams B, C, and D

(a) for team B

No.	M1	M2	M3
1	1-0	0-0	3-3
2	1-0	1-1	2-2
3	2-1	0-0	2-2
4	2-1	1-1	1-1
5	**3-2**	**0-0**	**1-1**
6	4-3	0-0	0-0

(b) for team C

No.	M1	M2	M3	No.	M1	M2	M3
1	3-0	0-0	2-3	7	4-1	1-1	0-1
2	3-0	1-1	1-2	8	5-0	0-0	0-3
3	3-0	2-2	0-1	9	5-1	0-0	0-2
4	4-0	0-0	1-3	10	5-2	0-0	0-1
5	**4-0**	**1-1**	**0-2**				
6	4-1	0-0	1-2				

(c) for team D

No.	M1	M2	M3
1	**0-0**	**0-1**	**0-4**
2	0-0	0-2	0-3

The scores for team B is given in Table 12.a, where there are a total of 10 scores, including 4 symmetrical cases, of which an appropriate one must selected such that $B \cap A = \{0-1\} \vee \{0-2\} \vee \{3-2\}$. Since team B has not lost, the only selection will be a score of $3-2$. The next step works for team C which can have the scores given in Table 12.b, where there are no symmetrical cases. A proper selection for team C might be made by using the sets $\{(0-1),(0-2)\}$ and $\{(0-0),(1-1)\}$ which are derived from the selected scores of teams A and B above, leaving the matched one out of the two sets, too. The cartesian product of these two sets produces four subsets

$$\{(0-1),(0-0)\} \quad \{(0-1),(1-1)\} \quad \{(0-2),(0-0)\} \quad \{(0-2),(1-1)\}.$$

which are searched among possible scores of team C. The aim here is to find a set of scores for team C such that $C \cap (A \cup B) = \{a,d\} \vee \{a,e\} \vee \{c,d\} \vee \{c,e\}$. Through the partitions in Table 12.b, the fifth one gives correct scores, that is $\{4-0, 1-1, 0-2\}$.

The scores of the last team, D, are generally used for the validity of the ones of the other teams. In this way, if the selections made for teams A, B and C are checked for team D, the first selection in Table 12.c will be the possible score of team D (Note that the scores for the solution are shown as bold in Table 10 and Table 12).

As a result, the total number of cases required to be examined via the blind search is the product of possible selections of scores for the teams. That is, there are $N(A) * N(B) * N(C) * N(D) = 13 * 6 * 10 * 2 = 1560$ cases above. According to the proposed vertex chaining approach, the number of cases examined is less.

7. Experimental Results

The primary goal of our experiments is to measure the time required to compute all possible scores with respect to both the number of matches and the FA value (i.e., the pair of $F - A$) in the tournament table. It is not considered which of the computed scores is meaningful for a football tournament. The measurements are carried out for only one team. Since the number of matches played in the tournament is directly associated with the number of teams, similar values for each team tend to occur.

Table 13. Number of the scores and time values for an average of 2.5 goals per match

Number of Matches	WDL	F-A	Number of Scores	Time
3	201	8-7	110	0 sec
4	202	10-9	918	0 sec
5	212	13-12	10256	3 sec
6	312	15-14	61132	15 sec
7	313	18-17	716796	13 min
8	413	20-19	3560226	64 min
9	414	23-22	34098562	6 days

Table 14. Number of the scores and time values for the fixed values of F and A

Number of Matches	WDL	F-A	Number of Scores	Time
3	201	20-19	3685	1 sec
4	202	20-19	51381	29 sec
5	212	20-19	177210	67 min
6	312	20-19	657530	4 min
7	313	20-19	2062412	39 min
8	413	20-19	3560226	64 min
9	414	20-19	5455645	25 hours

All the results presented below were obtained on Sun Solaris Sparc machine with 1.28 GHz each, 4 GB of RAM and 4 Ultra 160 SCSI hard disks with 73 GB each. For an average of 2.5 goals per match, Table 13 shows the number of possible scores and the time necessary to compute them, with a one-goal difference ($F - A = 1$) in total. The computations for the WDL values given in Table 13 lead to a maximum number of scores. The same computations are repeated for the fixed values of F and A, as seen in Table 14.

The other results re given for 3 matches in Table 15 and Table 16. The former shows the number of the scores with respect to different WDL and FA values. The latter gives the possible scores for the WDL value of 111 and the FA value of $4-3$, totaling to 10.

TABLE 15. Number of scores with respect to different WDL and FA values

F-A\WDL	102	111	120	201
1-0	–	–	1	–
2-1	–	1	2	1
3-2	1	4	4	3
4-3	3	**10**	6	9
5-4	9	20	9	19
6-5	19	35	12	38
7-6	38	56	16	66
8-7	66	84	20	110
9-8	110	120	25	170
10-9	170	165	30	255

TABLE 16. Possible scores for WDL value of 111 and FA value of $4-3$

No.	WS	DS	LS
1	2-0	2-2	0-1
2	2-0	1-1	1-2
3	2-0	0-0	2-3
4	3-0	1-1	0-2
5	3-0	0-0	1-3
6	3-1	1-1	0-1
7	3-1	0-0	1-2
8	4-0	0-0	0-3
9	4-1	0-0	0-2
10	4-2	0-0	0-1

To disclose the increase in the number of scores for various WDL and FA values that belong to 6 matches, the results are also displayed graphically. Fig.5 and Fig.6 shows the total number of the scores, keeping F and the goal difference fixed, correspondingly. They clearly demonstrates that the total number of possible scores gets larger as the goal difference decreases. Even for a fixed number of goal difference, the same occurs as the F and A values increase simultaneously. Besides, Fig.7 reflects a lower-increasing number of the scores for some number of matches which all ended with draws.

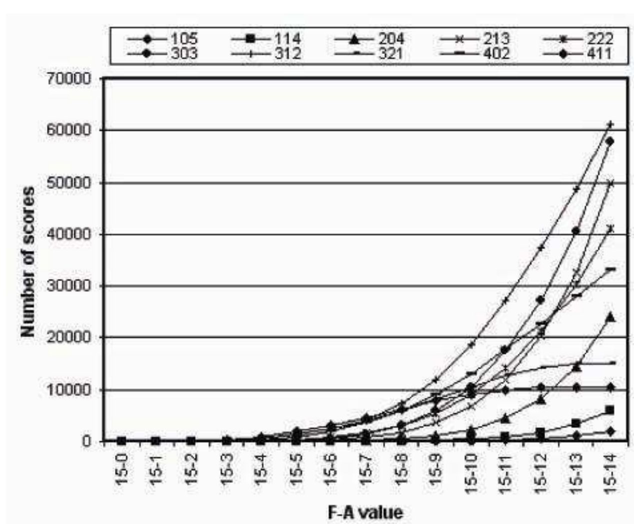

FIGURE 5. Increase in number of scores with decrease in goal difference

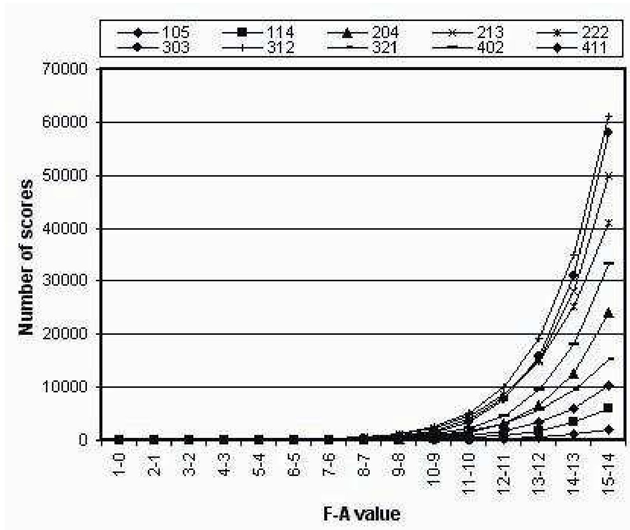

FIGURE 6. Increase in number of scores with increase in F and A

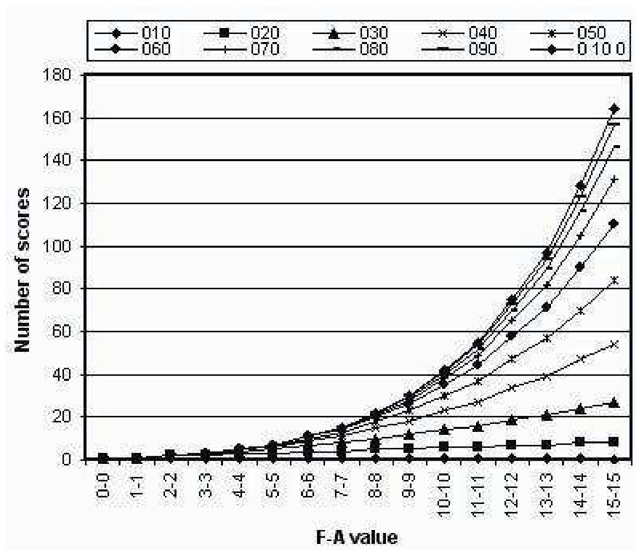

FIGURE 7. Increase in number of scores with $F = A$ for only draws

8. Conclusions

In this paper, we have focused on the determination of possible scores conforming to a given final table of a football tournament. Instead of the other blind search methods such as branch-and-bound, a rule-based solution strategy is developed, and applied to the problem. Since the solution of the problem, whose input data are the final standings of teams contained in the tournament table, handles a set of scores, it can be considered as a conditional partition problem. The solution comes in two stages.

- determination of the result of each particular match as 1, 0 or 2
- computation of the score of the match

Here, the result of the match is taken from the knowledge base under the given rules. The score is reached with two partial solutions. First the number of goals scored and conceded must be partitioned according to the number of played matches as well as the number of wins, draws, and losses. Then, using the proposed vertex chaining approach, the rule-bound solution of the problem is carried out, which leads to a regular black&white graph.

The proposed algorithm can easily be adapted to a sports league or other sports disciplines. From the perspective of teams, it has some roadmap facilities. For example, some sports teams may wish to analyse their current positions and find out which scores the games must result in to achieve a desired final position. The algorithm can be used to determine the required scores of the remaining games, according to the current standing of a participant in the competition. The same could also be done at the beginning of the tournament.

References

[1] Armstrong J, Willis RJ. Scheduling the cricket World Cup - a case-study. Journal of the Operational Research Society 1993;**44**; 1067-1072

[2] Bartsch T, Drexl A, Kröger S. Scheduling the professional soccer leagues of Austria and Germany. Computers and Operations Research 2006;**33**; 1907-1937

[3] Costa D. An evolutionary tabu search algorithm and the NHL scheduling problem. Information Systems and Operational Research 1995;**33**; 161-178

[4] Croce FD, Tadei R, Asioli PS. Scheduling a round robin tennis tournament under courts and players availability constraints. Annals of Operations Research 1999;**92**; 349-361

[5] Dejonghe T. Restructuring the Belgian professional football league: a location-allocation solution. Tijdschrift voor Economische en Sociale Geografie 2004;**95**; 73-88

[6] Easton K. Using integer programming and constraint programming to solve sports scheduling problems. PhD thesis, Georgia Institute of Technology, Atlanta; 2002.

[7] Fleurent C, Ferland JA. Allocating games for the NHL using integer programming. Operations Research 1993;**41**(4); 649-654

[8] Hart PE, Nilsson NJ, Raphael B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics SSC4 1968;**2**; 100–107

[9] Henz M. Scheduling a major college basketball conference-revisited. Operations Research 2001;**49**; 163-168

[10] Henz M, Mller T, Thiel, S. Global Constraints for Round Robin Tournament Scheduling. European Journal of Operational Research (EJORS) 2004;**153**(1); 92-101

[11] Kaski P. Algorithms for classification of combinatorial objects. Research Reports HUT-TCS-A94. Helsinki University of Technology Laboratory, Espoo, Finland; 1994.

[12] Lam CWH, Thiel L, Swiercz S. The non-existence of finite projective planes of order 10. Canadian Journal of Mathematics 1989;**41**; 1117-1123

[13] Linial N. Hard enumeration problems in geometry and combinatorics. SIAM Algebraic and Discrete Methods 1986;**7**; 331-335

[14] Rasmussen RV. Scheduling a triple round robin tournament for the best Danish soccer league. Technical report no. 2006/1, Department of OR, University of Aarhus, Denmark; 2006

[15] Rasmussen RV, Trick MA. Round robin scheduling - a survey. European Journal of Operational Research 2008;**188**(3); 617-636

[16] Redelmeier DH. Counting polyominoes: yet another attack, Discrete Mathematics 1981;**36**(2); 191-203

[17] Trick MA. Integer and constraint programming approaches for round-robin tournament scheduling. The 4th international conference on Practice and Theory of Automated Timetabling (PATAT 2002), Lecture Notes in Computer Science 2740, Springer, 2003; 63-77

[18] Wright M. Timetabling county cricket fixtures using a form of tabu search. Journal of the Operational Research Society 1994;**45**; 758-770

Vasif Nabiyev - was born in Azerbaijan. He received BSc. and MSc. degrees in the Faculty of Computing Technologies and Informatics from St. Petersburg Electrotechnical University in 1985, and a Ph.D. degree in the Department of Computer Science from Moscow Technical University in 1990. Then, in 2005, he received Professor degree in Computer Science Department from Karadeniz Technical University, Turkey, where he still lectures. He is currently the Chief of the Laboratory of Artificial Intelligence in Karadeniz Technical University. His research interests are in artificial intelligence, biometry, security, human-computer interaction, image processing, natural language processing, operational research, discrete and applied mathematics, combinatorial algorithms, game theory, logic programming.

He has published over 70 research papers, and authored three university level textbooks titled "Artificial Intelligence: "Problems, Methods and Algorithms" (1st ed. 2003, 2nd ed. 2005), "Introduction of Algorithms" (2007) and, "Combinatorial Algorithms" (2007).

Huseyin Pehlivan - was born in Turkey. He received a BSc. degree in Electrical and Electronics Engineering from Karadeniz Technical University, and MSc. and Ph.D. degrees in the Department of Computer Science from the University of Bristol. He is currently an assistant professor in the Department of Computer Engineering in Karadeniz Technical University. His research interests are broadly in systems security, software engineering, automated translation, functional programming and operational research. More precisely, he works on recovery systems, secure coding, identity theft, sports scheduling.