http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 13 (15): 2411-2417, 2014
ISSN 1812-5638 / DOI: 10.3923/itj.2014.2411.2417

Scheduling Volleyball Games using Linear Programming and Genetic Algorithm

[1,2]Fan-Wu Meng, [1]Ke-Chou Chen, [3]Kuo-Chuan Lin and [4]Rong-Chang Chen
[1]Office of Physical Education, Chung Yuan Christian University, Taiwan
[2]Graduate Institute of Sports Training, University of Taipei, Taiwan
[3]Graduate Institute of Athletics and Coaching Science, National Taiwan Sport University, Taiwan
[4]Department of Distribution Management,
National Taichung University of Science and Technology, Taiwan

Abstract: The purpose of this study is to employ a systematic approach to optimize the scheduling of volleyball tournaments. Given the total number of teams in a tournament, the number of game days, the number of courts and the number of teams at each division and the required time blocks at each game day can be optimally obtained by using integer programming. In addition, the referees can be optimally assigned to their preferred times by using Genetic Algorithm (GA). Results from the experiments show that the proposed approach can produce good solutions efficiently.

Key words: Genetic algorithm, integer programming, referee assignment, volleyball

INTRODUCTION

Volleyball is one of the most popular sports in the world (Briner and Kacmar, 1997; Ding, 2013). Since there is no physical contact between teams, volleyball is generally considered a moderate sport. It is also a team sport (Peeri *et al*., 2013) that can help players learn teamwork, experience team camaraderie and benefit their bodies. Many schools, therefore, hold a volleyball tournament to get students together both to boost the development of this sport and to enhance the physical and psychological health of students. A key factor to a successful volleyball tournament is the game scheduling (Kendall *et al*., 2010) which usually requires much time to plan and thus requires a systematic and efficient method to complete.

To optimize volleyball game scheduling, some problems should be figured out. First, the scheduling must satisfy some basic requirements. For example, given the total number of teams in a tournament, the game scheduler needs to optimally decide the number of teams at each division and the number of time blocks to arrange matches. The solutions are dependent on the number of game days and the number of courts. In addition, the scheduler must use a good scheme to arrange referees. One of the most complex issues in scheduling matches is the arrangement of referees. The number of referees required at each game is different for different tournament formats. Furthermore, the available time for each referee is different. Thus, a suitable scheme to arrange the times of referees is greatly needed. In this study, we will employ some suitable tools to deal with the problems mentioned above.

MATERIALS AND METHODS

Division optimization problem: Given the total number of teams participating in the tournament, organizers must decide how many teams to have at each division and the time blocks needed to complete the tournament. For example, if the total number of matches in the tournament is 66, the number of courts is three and the game days are six, then the time blocks per game day needed to complete the tournament will be not less than $66/(3\times6) = 3.67$. Thus, a good solution to the number of time blocks is four. Since the total number of matches depends on the tournament format and there are a variety of formats in practice, a systematic approach that can optimize the organizer's objectives is greatly needed.

Let, $G = \{1, 2,..., N_{division}\}$ be a set of divisions and $k\in G$. The required minimum number of teams at each division is designated as N_{min}, the allowed maximum number of teams at each division is N_{max}, the total number of matches is N_{match}, the total number of teams in the tournament is N_{team}, the number of game days is N_{day} and the number of courts is N_{court}. There are two decision variables: y_k and N_{block}, where y_k is the number of teams at division k and N_{block} is the number of time blocks on each game day. The problem can be represented mathematically as:

Corresponding Author: Rong-Chang Chen, Department of Distribution Management,
National Taichung University of Science and Technology, Taiwan

$$\text{Minimize } Z_1 = \text{Max}(y_k)\text{-Min}(y_k) \quad (1)$$

$$\text{Minimize } Z_2 = N_{block} \quad (2)$$

$$\sum_{k \in G} y_k = N_{team} \quad (3)$$

$$N_{min} \le y_k \le N_{max} \quad (4)$$

$$N_{min,\ block}, \le N_{block} \le N_{max,\ block} \quad (5)$$

$$N_{day} N_{block} N_{court} \ge N_{match} \quad (6)$$

$$y_k, N_{block} \in \text{Integer} \quad (7)$$

The first objective, Z_1 is to minimize the difference in the number of teams between divisions, as shown in Eq. 1. For example, if the total number of teams $N_{team} = 8$ and the number of divisions is three, Since, 8 = 1+1+6 = 1+2+5 = 1+3+4 = 2+2+4 = 2+3 +3, there are five possible solutions. The objective requires the difference of team numbers between divisions to be minimal. Thus, 2+3+3 is the best solution. This objective, Z_1 gives a more balanced tournament format. The second objective, Z_2 is to minimize the number of time blocks per game day. An apparent benefit to a smaller number of time blocks is the flexibility for game postponement due to rain or some other unexpected reasons.

There are some constrains in this problem. The number of teams at a division y_k is required to be no fewer than N_{min} and no greater than N_{max}, as illustrated in Eq. 4. Similarly, the number of time blocks is required to be no fewer than $N_{min,\ block}$ and no greater than $N_{max,\ block}$ as shown in Eq. 5. In addition, Eq. 6 requires that the number of matches to be arranged must be enough in given game days, court number and time block number. Equation 7 requires that y_k and N_{block} must be an integer.

Referee assignment problem: The referee assignment problem is a problem of arranging N_{block} time blocks to r referees. The preferences (or choices) of referees to time blocks are considered. The referees can set a maximum number of choices that they are available to play as referees. Each referee can choose at most N_p preferred time blocks. In this study, constraints that needed to be satisfied include:

- Each referee should referee at least M_{min} and at most M_{max} matches
- Each game requires exactly n referees

To formulate the problem mentioned above, some variables must be defined. Let, R = {1, 2,..., r} be a set of referees, T = {1, 2,..., N_{block}} be a set of time blocks and C = {1, 2,..., N_p} be a set of choices (preferences). For $i \in R$, $j \in T$, we can define p_{ij} as the preference given by referee i to being assigned time block j. This referee-time preference matrix ($r \times N_{block}$) contains the referees' preferences indicated by a value system of one standing for a first choice, two standing for a second choice and so on, up to a possible maximum value N_p, where the value of N_p is typically between five and ten. If referee i does not include block j in his/her list of preferences, then p_{ij} is assigned a suitably large penalty value Big. Furthermore, we may assign priority weights w_i to each referee, in order to give some referees a better chance of being assigned their preferred time blocks. The mathematical formulation of this problem can, therefore, be represented as:

$$\text{Minimize } F = \sum_{i \in R} \sum_{j \in T} w_i f(p_{ij}) x_{ij} \quad (8)$$

$$M_{min} \le \sum_{j \in T} x_{ij} \le M_{max} \quad \forall i \in R \quad (9)$$

$$\sum_{i \in R} x_{ij} = nN_{court} \quad \forall j \in T \quad (10)$$

$$x_{ij} = \begin{cases} 1 & \text{if referee i is ssigned to time block j,} \\ 0 & \text{othrewise} \end{cases} \quad (11)$$

$$\forall i \in R,\ j \in T \quad c_{ij} = \text{Big if } p_{ij} \notin C \quad (12)$$

The objective is to minimize the total score value of preferences, as shown in Eq. 8. Since a lower score value stands for a higher preference, the minimization of the total score value means the maximization of preference. Note that the function f (.) can be expressed in many forms. A typical function to be most used is the squared function (Chen, 2012; Chen *et al*., 2009, 2011; Harper *et al*., 2005). Equation 9 is used to limit the total number of times of refereeing for each referee, whereas, Eq. 10 requires that the number of referees at a time block is equal to n N_{court}. Eq. 11 requires that x_{ij} is either 1 or 0.

The referee assignment problem to be dealt with belongs to one of the generalized assignment problems (Cattrysse and van Wassenhove, 1992; Irving and Manlove, 2002; Liang *et al*., 2008; Lorena and Narciso, 1996; Martello and Toth, 1990; Narciso *et al*., 1999; Ross and Soland, 1975; Oncan, 2007). They are commonly addressed with heuristic algorithms. The generalized assignment problem is NP-hard (Harper *et al*., 2005). To solve problems of this type, Genetic Algorithm (GA) (Coley, 1999; Gen and Cheng, 1996, 2000; Goldberg, 1989; Holland, 1975; Huang *et al*., 2010; Kao *et al*., 2013; Mitchell, 1996; Winter *et al*., 1996) is one of the most

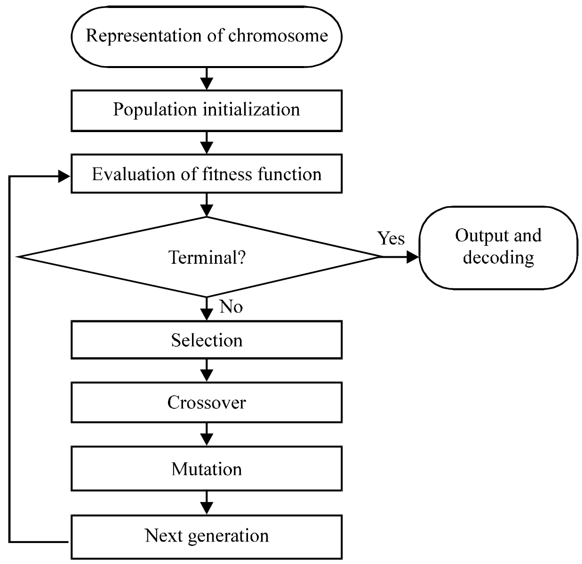

Fig. 1: Solution procedure in GA

effective algorithms which can obtain feasible solutions. Many previous studies employed GA to solve assignment problems and their results show that GA is effective (Chen and Hu, 2014; Chen and Hung, 2014; Chen *et al*., 2009; Chen *et al*., 2011; Harper *et al*., 2005). Therefore, in this study, we will employ GA to solve the problem as stated.

Method of solution: To decide on the number of teams at each division, an integer programming tool is used. On the other hand, the referee assignment problem is solved by using GA. The procedure to perform GA is illustrated in Fig. 1 (Chen *et al*., 2011).

Encoding: In GA, one of the most important operations is the encoding of a chromosome. If the number of referees at each time block is one, the method of encoding is represented as Fig. 2. The value of a gene stands for the referee assigned to the time block. For instance, the value of the first gene in the first time block is 7, showing that referee 7 is assigned to time block 1.

If the number of referees at each block is more than one, say m, then the number of values at each gene is equal to m. For example, if m = 2, the encoding can be represented as shown in Fig. 3.

Initial population and evaluation: The initial population is generated using a random scheme, i.e., the values of genes are given randomly. After genetic operations, we need to evaluate the performance of the chromosomes according to the referees' preferences. For example, if

Time block	1	2	3	4	5	6	...	8
Referee	7	2	8	3	5	10	...	11

Fig. 2: Representation of a chromosome for single referee

Time block	1	2	3	4	5	6	...	8
Referee	3,5	4,7	3,8	2,9	5,10	1,10	...	7,8

Fig. 3: Representation of a chromosome for two referees

there are three referees and they are assigned 1st, 3rd and 5th choice time blocks, the scores are 1^2, 3^2 and 5^2, respectively. Then the fitness value equals $1^2+3^2+5^2 = 35$.

Selection, crossover and mutation: We adopt the binary tournament selection method to choose the parent chromosomes from the population. The process is as follows: First, two chromosomes are chosen randomly from the population. The GA program compares the chromosomes according to their fitness values and keeps the better one to become parent A. The GA program then repeats the process to select parent B. After that, the GA program combines parents A and B to produce offspring. In this study, the crossover is done by a single-point crossover scheme. A point is randomly selected and parent chromosomes are divided into two parts. The gene values at the first part of parent chromosome A and the gene values at the second part of parent chromosome B are copied, to generate a child chromosome. At the same time, the gene values at the second part of parent chromosome A and at the first part of parent chromosome B are copied, to generate the other child chromosome.

The mutation method we adopt is the change method. Two genes in a chromosome are randomly selected and their values are exchanged, to generate a new chromosome.

Termination: The GA program is run until a pre-assigned generation number is reached. Then the assignment results are output.

RESULTS AND DISCUSSION

The GA program was developed using VB.NET 2008. All the experiments were tested on a PC with an AMD Athlon (tm) 2.6 GHz 64 Processor and 512 RAM. To validate the GA program, Integer Programming (IP) was also performed.

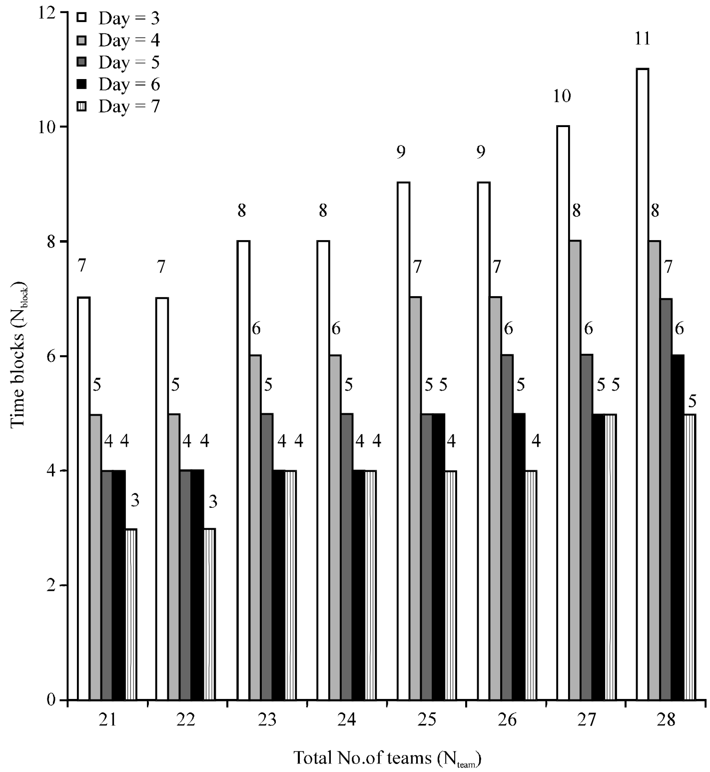

Fig. 4: Variation of the number of time blocks (N_{block}) with the total number of teams (N_{team}) and the number of game days (N_{day}). The number of courts (N_{court}) is 2

To investigate the effectiveness of the proposed approach, a base case was setup with the following settings; No. of teams, N_{team} = 26, No. of courts, N_{court} = 3, No. of game days, N_{day} = 6, No. of divisions, $N_{division}$ = 6, the minimum No. of teams per division, N_{min} = 3 and the maximum No. of teams per division, N_{max} = 5. The number of referees is 45. The minimum number of times to judge a match, N_{min} = 4 and the maximum number of times to judge a match, N_{max} = 6. The tournament format is as follows:

Preliminaries: Round robin pool play; the best two teams advance to the second round.

Second round: There are two divisions. Each division employs the round robin competition. Best two teams go to the semi-finals.

Semi-finals: There are two matches. The winners advance to the finals. The losers play for the third place.

Finals: There are two matches which decide the champion, the second place, the third place and the fourth place.

Decision of y_k and N_{block}: The number of teams at each division can be obtained by integer programming. The values of y_k = 4, 4, 4, 4, 5 and 5. Since the preliminaries use the round robin competition, the total number of matches should be equal to $C_2^4+C_2^4+C_2^4+C_2^4+C_2^5+C_2^5$ = 6+6+6+6+10+10 = 44. As for the second round, the number of matches = $C_2^3+C_2^3$ = 3+3 = 6. Thus, the total number of matches N_{match} = 44+6+2+2 = 54.

When the tournament format is known, the values of y_k and N_{block} depend on the total number of teams in the tournament, N_{team}, the number of courts, N_{court} and the number of game days, N_{day}. Table 1 summarizes the solutions at different N_{team}, N_{court} and N_{day}.

The variation of the number of time blocks, N_{block} with different N_{team}, N_{day} and N_{court} is shown in Fig. 4 and 5. In Fig. 4, the number of courts, N_{court} is two while the number of courts N_{court} is three in Fig. 5. As we can see from these two figures, the influences of the number of game days, N_{day} and the number of courts, N_{court} on the number of time blocks N_{block} are quite significant.

Referee assignment results: At the base case, the total number of matches (N_{match}) is 54. Since the number of time blocks (N_{block}) is found to be 3 and the number of game days (N_{day}) 6 there are 18 time blocks in total which are represented as TB01, TB02,..., TB18. If every match needs four referees, the total number of referees needed is equal to 54×4 = 216. Solving the problem by GA program, the assignment results are illustrated in Fig. 6a and b. Good

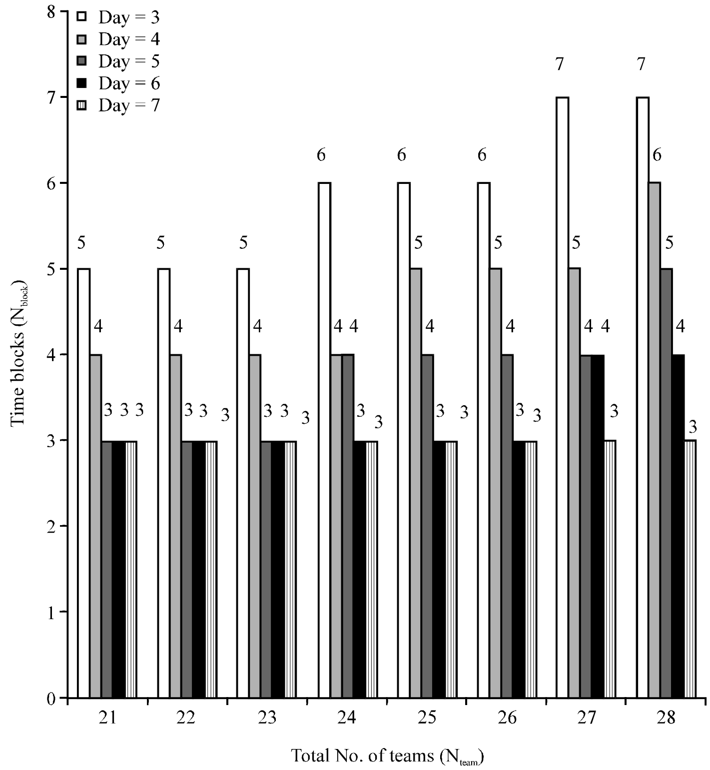

Fig. 5: Variation of the number of time blocks (N_{block}) with the total number of teams (N_{team}) and the number of game days (N_{day}). The number of courts (N_{court}) is 3

(a)

Time/referee	Blocks											
TB01	R004	R004	R004	R007	R005	R006	R017	R019	R020	R021	R031	R040
TB02	R002	R002	R002	R001	R003	R004	R012	R015	R016	R024	R034	R043
TB03	R026	R026	R026	R009	R010	R016	R018	R018	R013	R028	R037	R043
TB04	R006	R006	R006	R012	R017	R022	R002	R008	R013	R017	R038	R016
TB05	R038	R038	R038	R013	R014	R025	R027	R001	R010	R012	R015	R009
TB06	R003	R021	R021	R031	R033	R034	R040	R006	R013	R018	R045	R017
TB07	R003	R017	R017	R008	R020	R001	R006	R026	R036	R042	R045	R016
TB08	R001	R005	R034	R023	R016	R018	R004	R012	R014	R017	R019	R020
TB09	R024	R007	R009	R027	R029	R038	R002	R010	R015	R020	R022	R041
TB10	R033	R042	R004	R011	R019	R024	R043	R044	R003	R005	R023	R018
TB11	R015	R024	R034	R002	R022	R025	R026	R032	R035	R036	R041	R045
TB12	R010	R006	R026	R028	R030	R007	R009	R011	R018	R029	R038	R042
TB13	R019	R021	R031	R032	R035	R044	R003	R023	R030	R033	R039	R042
TB14	R013	R014	R022	R028	R030	R032	R037	R039	R041	R001	R011	R021
TB15	R008	R011	R014	R016	R023	R031	R032	R033	R040	R010	R044	R017
TB16	R012	R020	R021	R030	R039	R029	R036	R038	R045	R014	R027	R019
TB17	R019	R029	R044	R037	R043	R005	R020	R025	R035	R008	R039	R040
TB18	R015	R018	R027	R028	R036	R037	R045	R004	R039	R040	R040	R044

Fig. 6(a-b): Continue

(b)

Referee/time	TB01	TB02	TB03	TB04	TB05	TB06	TB07	TB08	TB09	TB10	TB11	TB12	TB13	TB14	TB15	TB16	TB17	TB18
R001		2(TB02)			3(TB05)		4(TB07)	1(TB08)						3(TB14)				
R002		1(TB02)	2(TB03)	4(TB04)					5(TB09)		3(TB11)							
R003		2(TB02)				4(TB06)	1(TB07)			5(TB10)			3(TB13)					
R004	1(TB01)	3(TB02)						5(TB08)		4(TB10)								2(TB18)
R005	3(TB01)		2(TB03)					1(TB08)		5(TB10)							4(TB17)	
R006	3(TB01)			1(TB04)		5(TB06)	4(TB07)					2(TB12)						
R007	2(TB01)	1(TB02)		5(TB04)					4(TB09)			3(TB12)						
R008			3(TB03)	4(TB04)			2(TB07)								1(TB15)		5(TB17)	
R009	1(TB01)		2(TB03)		5(TB05)				4(TB09)			3(TB12)						
R010			2(TB03)		4(TB05)				5(TB09)			1(TB12)			1(TB15)			
R011					2(TB05)					4(TB10)		3(TB12)		5(TB14)	1(TB15)			
R012		3(TB02)		2(TB04)	4(TB05)			5(TB08)								1(TB16)		
R013			3(TB03)	4(TB04)	2(TB05)	5(TB06)								1(TB14)				
R014					2(TB05)			5(TB08)						4(TB14)	1(TB15)	3(TB16)		
R015		3(TB02)			4(TB05)				5(TB09)		2(TB11)							1(TB18)
R016		3(TB02)	2(TB03)	6(TB03)			5(TB07)	4(TB08)							1(TB15)			
R017	3(TB01)			2(TB03)		6(TB06)	1(TB07)	5(TB08)							4(TB15)			
R018			2(TB03)			5(TB06)		4(TB08)		6(TB10)		3(TB12)						1(TB18)
R019	3(TB01)							5(TB08)		4(TB10)			2(TB13)			7(TB16)	1(TB17)	
R020	3(TB01)						2(TB07)	6(TB08)	5(TB09)							1(TB16)	4(TB17)	
R021	3(TB01)					4(TB06)							2(TB13)	5(TB14)		1(TB16)		
R022		2(TB02)		4(TB04)					5(TB09)		3(TB11)			4(TB14)				
R023						4(TB06)		2(TB08)		5(TB10)			3(TB13)		1(TB15)			
R024		3(TB02)							1(TB09)	4(TB10)	2(TB11)							
R025					2(TB05)		1(TB07)				3(TB11)						4(TB17)	
R026			1(TB03)				4(TB07)				3(TB11)							
R027					2(TB05)				4(TB09)							3(TB16)		1(TB18)
R028			3(TB03)									2(TB12)		4(TB14)				1(TB18)
R029									4(TB09)			3(TB12)				2(TB16)	1(TB17)	
R030												2(TB12)	3(TB13)	4(TB14)		1(TB16)		
R031	3(TB01)					4(TB06)							2(TB13)		1(TB15)			
R032											3(TB11)		3(TB13)	4(TB14)	1(TB15)			
R033						4(TB06)				2(TB10)			3(TB13)		1(TB15)			
R034		3(TB02)				4(TB06)		1(TB08)			2(TB11)							
R035				1(TB04)							3(TB11)		2(TB13)				4(TB17)	
R036							4(TB07)				3(TB11)					2(TB16)		1(TB18)
R037			3(TB03)											4(TB14)			2(TB17)	1(TB18)
R038				5(TB04)	1(TB05)				4(TB09)			3(TB12)				2(TB16)		
R039													3(TB13)	4(TB14)		1(TB16)	5(TB17)	2(TB18)
R040	3(TB01)					4(TB06)									1(TB15)		5(TB17)	2(TB18)
R041	1(TB01)								5(TB09)		3(TB11)			4(TB14)				2(TB18)
R042					1(TB05)		4(TB07)			2(TB10)		5(TB12)	3(TB13)					
R043		3(TB02)	5(TB03)	1(TB04)						4(TB10)							2(TB17)	
R044										4(TB10)			2(TB13)		3(TB15)		1(TB17)	5(TB18)
R045						5(TB06)	4(TB07)				3(TB11)					2(TB16)		1(TB18)

Fig. 6(a-b): (a) Referee assignment results for the base case and (b) Detailed referee assignment results for the base case

Table 1: Solutions at different total numbers of teams

N_{team}	y_k	N_{match}
21	3,3,3,4,4,4	37
22	3,3,4,4,4,4	40
23	3,4,4,4,4,4	43
24	4,4,4,4,4,4	46
25	4,4,4,4,4,5	50
*26	4,4,4,4,5,5	54
27	4,4,4,5,5,5	58
28	4,4,5,5,5,5	62

*Base case

results can be obtained by GA in just a few seconds and this shows that the GA approach is quite efficient. In addition, the result satisfies the constraints of the problem, demonstrating that the proposed approach is effective.

CONCLUSION

To boost the development of volleyball, a variety of tournaments are held continually in many schools. A key factor to a successful volleyball tournament is the game scheduling which typically requires much time to design and thus needs an efficient method to complete. In this study, we employ a systematic approach to optimize the volleyball game scheduling. Given the total number of teams participating in the tournament, the number of game days expected to complete the tournament and the number of courts the venue can provide, the number of teams at each division and the required time blocks on each game day can be optimally obtained by the integer programming technique. In addition, the referees can be optimally assigned to their preferred times by using genetic algorithm. Results from the experiments showed that the proposed approach can reach good solutions efficiently.

ACKNOWLEDGMENTS

The authors wish to express their appreciation to Jheng-Bang Chen and Nancy D. Wall for their help during the course of this study. This study was supported by the National Science Council under grant Numbers NSC 102-2221-E-025-013 and 98-2815-C-025 -003-E.

REFERENCES

Briner Jr., W.W. and L. Kacmar, 1997. Common injuries in volleyball. Sports Med., 24: 65-71.

Cattrysse, D.G. and L.N. van Wassenhove, 1992. A survey of algorithms for the generalized assignment problem. Eur. J. Operat. Res., 60: 260-272.

Chen, R.C. and P.H. Hung, 2014. Multi objective order assignment optimization in a Global multiple-factory environment. Math. Prob. Eng., Vol. 2014. 10.1155/2014/673209

Chen, R.C. and T.T. Hu, 2014. A decision-making mechanism considering carbon footprint and cost to fulfill orders for multi-site global companies, to appear. Int. J. Shipping Transport Logistics (In Press).

Chen, R.C., 2012. Grouping optimization based on social relationships. Math. Prob. Eng., Vol. 2012. 10.1155/2012/170563

Chen, R.C., M.R. Huang, R.G. Chung and C.J. Hsu, 2011. Allocation of short-term jobs to unemployed citizens amid the global economic downturn using genetic algorithm. Exp. Syst. Appl., 38: 7535-7543.

Chen, R.C., T.T. Chen and W.L. Fang, 2009. Assignment of external off-the-job training courses to employees using genetic algorithm. Inform. Technol. J., 8: 147-155.

Coley, D.A., 1999. An Introduction to Genetic Algorithms for Scientists and Engineers. 1st Edn., World Scientific Press, Singapore.

Ding, J., 2013. Volleyball games prediction and evaluation model by homogeneous Markov chain based on fuzzy neural network. Int. J. Applied Math. Stat., 46: 404-412.

Gen, M. and R. Cheng, 1996. Genetic Algorithms and Engineering Design. 1st Edn., Wiley, New York.

Gen, M. and R. Cheng, 2000. Genetic Algorithms and Engineering Optimization. Wiley and Sons, New York, USA., ISBN-13: 9780471315315, Pages: 495.

Goldberg, D.E., 1989. Genetic Algorithm in Search, Optimization and Machine Learning. 1st Edn., Addison-Wesley, Reading, MA, USA.

Harper, P.R., V. de Senna, I.T. Vieira and A.K. Shahani, 2005. A genetic algorithm for the project assignment problem. Comput. Oper. Res., 32: 1255-1265.

Holland, J.H., 1975. Adaptation in Natural and Artificial Systems. 1st Edn., University of Michigan Press, Ann Arbor, Michigan, ISBN: 9780472084609, Pages: 183.

Huang, C.C., R.G. Chung, R.C. Chen, T.S. Chen, T.N. Le, C.J. Hsu and Y.C. Tsai, 2010. Finding an optimal combination of key training items using genetic algorithms and support vector machines. Inform. Technol. J., 9: 652-658.

Irving, R.W. and D.F. Manlove, 2002. The stable roommates problem with ties. J. Algorithms, 43: 85-105.

Kao, Y.H., T.S. Chen, W.B. Lee, R.C. Chen, C.C. Huang, M.C. Lin and Y.L. Wang, 2013. Key training items search of manufacturing assessment based on TTQS and GA-SVM. Inform. Technol. J., 12: 756-762.

Kendall, G., S. Knust, C.C. Ribeiro and S. Urrutia, 2010. Scheduling in sports: An annotated bibliography. Comput. Oper. Res., 37: 1-19.

Liang, X., R.C. Chen and X. Guo, 2008. Pruning support vector machines without altering performances. IEEE Trans. Neural Networks, 19: 1792-1803.

Lorena, L.A.N. and M.G. Narciso, 1996. Relaxation heuristics for a generalized assignment problem. Eur. J. Operat. Res., 91: 600-610.

Martello, S. and P. Toth, 1990. Knapsack Problems: Algorithms and Computer Implementation. John Wiley and Sons, New York, ISBN: 0-471-92420-2, pp: 296.

Mitchell, M., 1996. An Introduction to Genetic Algorithms. MIT Press, Cambridge, MA, USA., ISBN:0-262-13316-4.

Narciso, M.G., L. Antonio and N. Lorena, 1999. Lagrangean/surrogate relaxation for generalized assignment problems. Eur. J. Oper. Res., 114: 165-177.

Oncan, T., 2007. A survey of the generalized assignment problem and its applications. INFOR: Inform. Syst. Oper. Res., 45: 123-141.

Peeri, M., R. Sharif and H. Matinhomaee, 2013. Relations of some corporeal properties with performances of volleyball players who participated in Japan world competitions. Eur. J. Exp. Biol., 3: 88-94.

Ross, G.T. and R.M. Soland, 1975. A branch and bound algorithm for the generalized assignment problem. Math. Programm., 8: 91-103.

Winter, G., J. Periaux and M. Galan, 1996. Genetic Algorithms in Engineering and Computer Science. Wiley, New York.