Model Transformation and Optimization of the Olympics Scheduling Problem[§]

Yongheng Jiang*†, Qinghua Gu*, Biqing Huang*, Xi Chen*, Tianyuan Xiao*
*Department of Automation, Tsinghua University, Beijing 100084, PR China
†National Laboratory of Intelligent Technology and Systems, Beijing 100084, PR China

ABSTRACT Model transformation and optimization of the Olympics scheduling problem are studied in this paper. (1) The Olympics scheduling problem involves several classes of constraints, which key is the temporal interval. So, a temporal interval model language is presented to model the scheduling problem. (2) The Olympics scheduling problem is a constraint satisfaction problem. Considering its natural complexity, the constraint satisfaction problem is transformed into a constrained optimization problem by softening the time constraints of the final matches. (3) In the constrained optimization model, the events are coupled by the field constraints, by ignoring which, the events are independent to each other. So, the constrained optimization problem is decomposed into separable event sub-problems by relaxing the field constrains. The global optimal solutions could be obtained by adjusting the Lagrangian multipliers. (4) To adjust the Lagrangian multipliers, the subgradient projection method with variable diameter is presented, which converges without any priori knowledge, and the convergence efficiency is given. The numerical testing results show the convergence and the performance difference between the subgradient projection method with variable diameter and the reduced algorithm. The phase transition of the primal constraint satisfaction problem could be recognized by the number of tests to get q>0 and the time first to get q>0 (q is the dual value) generalized by the subgradient projection method with variable diameter.

Keyword: Scheduling; Olympics; Lagrangian Relaxation; Model transformation

1 Introduction

From 1980s, modern sports activities are more and more socialized, specialized, entertainized, commercialized and informatized. As time flies, new technologies play more and more important roles in sports organization. And “High-tech Olympics” has been one of the three themes of “New Beijing, Great Olympics”. Research progress on “High-tech Olympics” will benefit the society and economy development.

The problem of representing temporal knowledge and temporal reasoning is a core problem of the Olympics system designing and scheduling. Temporal information technology has experienced 3 history periods [7]. The inaugurating period before 1982, the Ph. D. dissertations of Ben-Zvi

[§] This research was partially funded by Beijing NSF Project under Grant 4031002.
[*] Corresponding author. Email: jiangyh@tsinghua.edu.cn. Tel: 00 86 10 62772704-822. Fax: 00 86 10 62786911

(California State University, Los Angeles) and Clifford (New York University) in 1982 are the representative results. In this period, the time interval was represented as a field value, and the bi-temporal of valid time and transaction was imported. The flourishing period is from 1983 to 1993. In this period, the theoretical model of temporal database had come into being. The third period starts in 1994, and the application of the temporal information is the important character. In 1983, Allen [8] studied the problem of representing temporal knowledge and temporal reasoning, and represented 13 allowable temporal relations which may hold between intervals. And Parthasarathy [9, 10] extended Allen's results. But Parthasarathy's results are helpless for some event relations in the Olympics such as decentralization constraints etc.

Olympics organization involves quite many sorts of resources, athletes, refree and gymnastic, as long with various constraints. So the Olympics scheduling is an arduous task, which is one of the important branches of the "High-tech Olympics" research. Sports scheduling research originates in the 1980s. Nemhauser and Trick[1], Henz et. al[2], Schaerf[3], Regin[4], McAloon and Tretkoff, Wetzel[5], Schönberger, Mattfeld and Kopfer[6] have explored the scheduling problem. But they focus on the single-event tournament scheduling problem, research on multi-event and large scaled scheduling research has not been reported.

The Olympics scheduling problem is a timetabling problem. The timetabling problem is a combination problem in nature, and has been proven NP-hard. A wide variety of approaches to timetabling problems have been described in the literature and tested on real data. They can be roughly divided into four types [11]: (1) sequential methods, (2) cluster methods, (3) constraint-based methods, and (4) meta-heuristic methods. Sequential methods and cluster methods present policies that obtain approximate solutions. Constraint-based methods are back-tracking methods in nature. Meta-heuristic methods could provide good solutions, but it is computation consumed.

A novel method might be found by transforming the problem into a new model. In this paper, the time constraints of the final matches are softened, and then the Olympics scheduling problem is transformed into a constrained optimization problem. Observing that the events are coupled by the field constraints, the constrained optimization problem can be decomposed into single-event sub-problems by relaxing the field constraints. Lagrangian relaxation provides a methodology to realize the decomposition. From 1990s, the authors such as Peter B. Luh and etc. have studied the production scheduling problem with Lagrangian relaxation [12-17]. They relaxed the capacity constraints to decompose the production scheduling problem into job sub-problems. The difficulty of Lagrangian relaxation is the optimization of the dual problem. In 1969, Polyak [18] has presented a subgradient method to the convex problem, but the convergence of the method depends on the estimated optimal value. In 1996, Kiwiel [19, 20] studied the subgradient projection methods for convex optimization, which converges without the estimated optimal value, but the diameter of the variable domain is imported. Kim [21] presented a variable target value subgradient method in 1991 before Kiwiel, which is nothing but a special case of Kiwiel's method.

In section 2, a temporal interval language is represented. In section 3, the Olympics scheduling problem is modeled, and the model is transformed into a constrained optimization problem. Then a solution methodology is presented based on Lagrangian ralxation in section 4, and the subgradient projection method with variable diameter is presented together with 3 lemmas and 1 theorem, which converges without any priori knowledge by a given efficiency. In Section 5, the numerical results are discussed. And a conclusion is drawn in section 6. The lemmas and theorem are proven in the appendix.

2 Temporal Interval Model Language

The Olympics system involves time-distribution constraints, field constraints, person constraints and time window constraints. The existing temporal interval model languages are not able to model all the constraints. So a new temporal interval model language is represented as follows.

The matches are scheduled in the periods of the competition days, and the periods are not continuous. So a triple (d, p, τ) is designed for time t, $d = 1, \cdots, n$ denotes the days, $p = 0,1,\cdots, p_m - 1$ denotes the periods (p_m is the number of the periods in a day), and $\tau = [0,1,\cdots,\tau_m - 1]$ denotes the time point (τ_m is the number of the time points in a period). In the Olympics scheduling problem, the k^{th} match of event i's j^{th} round is denoted as (i, j, k), and the final match is denoted as $(i, j_{mi}, 1)$. The beginning time and ending time of (i, j, k) is denoted as T_{bijk} and T_{eijk}, the corresponding triples are $(d_{bijk}, p_{bijk}, \tau_{bijk})$ and $(d_{eijk}, p_{eijk}, \tau_{eijk})$. The operation rules of the triple are as following.

1) addition

$$(d_1, p_1, \tau_1) + (d_2, p_2, \tau_2) = (d_3, p_3, \tau_3)$$

Where

$$\tau_3 = MOD(\tau_1 + \tau_2, \tau_m)$$

$$p_3 = MOD(p_1 + p_2 + INT((\tau_1 + \tau_2)/\tau_m), p_m)$$

$$d_3 = d_1 + d_2 + INT((p_1 + p_2 + INT((\tau_1 + \tau_2)/\tau_m)/p_m)$$

$MOD(a,b)$ is the remainder of a divided by b, and $INT(a)$ is the maximal integer smaller than a.

$$(d_1, p_1, \tau_1) + t = (d_2, p_2, \tau_2)$$

Where

$$\tau_2 = MOD(\tau_1 + t, \tau_m)$$

$$p_2 = MOD(p_1 + INT((\tau_1 + t)/\tau_m), p_m)$$

$$d_2 = d_1 + INT((p_1 + INT((\tau_1 + t)/\tau_m)/p_m)$$

$$d_2 = d_1 + INT((p_1 + INT((\tau_1 + t)/\tau_m)/p_m)$$

2) subtraction

Subtraction is the converse computation of addition.

For $(d_1, p_1, \tau_1) + (d_2, p_2, \tau_2) = (d_3, p_3, \tau_3)$, $(d_1, p_1, \tau_1) = (d_3, p_3, \tau_3) - (d_2, p_2, \tau_2)$;

For $(d_1, p_1, \tau_1) + t = (d_2, p_2, \tau_2)$, $(d_1, p_1, \tau_1) = (d_2, p_2, \tau_2) - t$.

3) compare

Equal

$$(d_1, p_1, \tau_1) = (d_2, p_2, \tau_2) \Leftrightarrow d_1 = d_2 \wedge p_1 = p_2 \wedge \tau_1 = \tau_2 .$$

Smaller

$$(d_1, p_1, \tau_1) < (d_2, p_2, \tau_2)$$

$$\Leftrightarrow d_1 < d_2 \vee d_1 = d_2 \wedge p_1 < p_2 \vee d_1 = d_2 \wedge p_1 = p_2 \wedge t_1 < t_2$$

Lager

$$(d_2, p_2, \tau_2) > (d_1, p_1, \tau_1) \Leftrightarrow (d_1, p_1, \tau_1) < (d_2, p_2, \tau_2)$$

Smaller or equal

$$(d_1, p_1, \tau_1) \le (d_2, p_2, \tau_2) \Leftrightarrow (d_1, p_1, \tau_1) < (d_2, p_2, \tau_2) \vee (d_1, p_1, \tau_1) = (d_2, p_2, \tau_2)$$

Lager or equal

$$(d_2, p_2, \tau_2) \ge (d_1, p_1, \tau_1) \Leftrightarrow (d_1, p_1, \tau_1) \le (d_2, p_2, \tau_2)$$

4) conversion between the triple difference and the length of time

$$L((\Delta d, \Delta p, \Delta \tau)) = (\Delta d \cdot p_m + \Delta p) \cdot \tau_m + \Delta \tau$$

Next, the time interval and its relations are represented.

$[t_1, t_2]$ denotes a time interval, and t_1 and t_2 may be a same one of the three forms as (d, p, τ), (d, p) and d.

$$[d_1, d_2] \Leftrightarrow [(d_1, 0), (d_2, p_m - 1)]$$

$$[(d_1, p_1), (d_2, p_2)] \Leftrightarrow [(d_1, p_1, 0), (d_2, p_2, \tau_m - 1)]$$

There are the following relations for time interval.

1) equal

$$[t_{11}, t_{12}] = [t_{21}, t_{22}] \Leftrightarrow t_{11} = t_{21} \wedge t_{12} = t_{22}$$

We say that one time interval is equal to the other if their left extreme points and right extreme points are equal respectively.

2) before

$$[t_{11}, t_{12}] < [t_{21}, t_{22}] \Leftrightarrow t_{12} < t_{21}$$

Time interval I is before II if I ends earlier than II begin.

3) after

$$[t_{11}, t_{12}] > [t_{21}, t_{22}] \Leftrightarrow t_{11} > t_{22}$$

Time interval I is after II if I begins later than II end.

4) repulsive

$$[t_{11}, t_{12}] \overset{c_1}{\underset{c_2}{\leftrightarrow}} [t_{21}, t_{22}] \Leftrightarrow t_{12} + c_1 < t_{21} \vee t_{11} > t_{22} + c_2$$

Time interval I is repulsive to II if I ends at least c_1 earlier than II begin, or begins at least c_2 later than II end.

5) close

$$[t_{11}, t_{12}] \overset{c_1}{\underset{c_2}{\approx}} [t_{21}, t_{22}] \Leftrightarrow t_{11} \le t_{22} + c_2 \wedge t_{12} + c_1 \ge t_{21}$$

Time interval I is close to II if I ends at least c_1 later than II begin, and begins at least c_2 earlier than II end.

6) including

$$[t_{11}, t_{12}] \supset [t_{21}, t_{22}] \Leftrightarrow t_{11} \le t_{21} \wedge t_{12} \ge t_{22}$$

Time interval I is including II if I begins earlier than II begin, and ends later than II end.

7) during

$$[t_{11}, t_{12}] \subset [t_{21}, t_{22}] \Leftrightarrow t_{11} \ge t_{21} \wedge t_{12} \le t_{22}$$

Time interval I is including II if I begins later than II begin, and ends earlier than II end.

The rules above are used to model the Olympics scheduling problem of a schedule system.

3 Modeling and Model Transforming of the Olympics Scheduling Problem

In Olympics, the athletes hope to play fairly, the audience wants their favorable matches, the sponsors require the gainful advertisement time, and the matches run in some special order.

For most events, teams or athletes take part in some matches, and the winners are promoted to next round, until the champion is chosen. There are one or more matches in a round, and the matches of the preceding round have to be earlier than those of the successor. These are called order constraints. Some special event has just one round. For some traditional or other reasons, a special round of event A must run before related round of event B and after the preceding round, for example, swimming man and swimming woman. These are called cross constraints. Some event cannot be much earlier or later than some other event. These are called close constraints. Some event has to be some time earlier or later than some other event. For example, the different groups of the same event have better to run one after another, and the track events longer than 1000 meters cannot run in a day. These are called decentralization constraints. All of the constraints before are called time-distribution constraints.

The capacity of the fields is limited, so there cannot be much more matches on the fields at the same time. Some fields may affect each other, and there cannot be matches on the fields in the same time. And these are called field constraints.

In Olympics, some athlete may attend more than one event. So there has to be some time interval between the correlative events, and the athlete cannot run from one field to another in a period of time. These are called person constraints.

The audience and sponsors' requires could be expressed by time window constraints.

In fact, the events of the Olympics are coupled by the field constraints. The field uncoupled events could be schedule respectively and independently. Next, we analyze and formulate a kind of field coupled events with the temporal interval model language represented in section 2.

First, we explain some symbols. If event i can run on the current kind of field, $S(i)=1$, otherwise, $S(i)=0$. If match i run on the m^{th} field, $O[(i,j,k),m,t]=1$, otherwise, $O[(i,j,k),m,t]=0$.

The constraints are formulated as follows:

1) Time-distribution constraints

Order constraints are as follows:

$$[T_{bi_1j_1k_1},T_{ei_1j_1k_1}]<[T_{bi_2j_2k_2},T_{ei_2j_2k_2}] \tag{1}$$

Match (i_1,j_1,k_1) must be before match (i_2,j_2,k_2).

Cross constraints are as follows:

$$[T_{bi_1jk_1},T_{ei_1jk_1}]<[T_{bi_2jk_2},T_{ei_2jk_2}]\wedge[T_{bi_2jk_2},T_{ei_2jk_2}]<[T_{bi_1j+1k_1},T_{ei_1j+1k_1}] \tag{2}$$

Event i_2's j^{th} round must be after event i_1's j^{th} round, and before event i_1's $j+1^{\text{th}}$ round.

Decentralization constraints are as follows:

$$[T_{bi_1j_1k_1},T_{ei_1j_1k_1}]\overset{c_1}{\underset{c_2}{\leftrightarrow}}[T_{bi_2j_2k_2},T_{ei_2j_2k_2}] \tag{3}$$

Match (i_1,j_1,k_1) must be repulsive to match (i_2,j_2,k_2).

Close constraints are as follows:

$$[T_{bi_1j_1k_1},T_{ei_1j_1k_1}]\overset{c_1}{\underset{c_2}{\approx}}[T_{bi_2j_2k_2},T_{ei_2j_2k_2}] \tag{4}$$

Match (i_1,j_1,k_1) must be close to match (i_2,j_2,k_2).

2) Field constraints

$$\forall l,t\ ,\quad \sum_{i:S(i)=1} O[(i,j,k),m,t]\le m_{\max} \tag{5}$$

There cannot be more than $m_{\max}$ matches on the fields at the same time.

3) Person constraints

$$[T_{bi_1j_1k_1}, T_{ei_1j_1k_1}] \underset{c_2}{\overset{c_1}{\leftrightarrow}} [T_{bi_2j_2k_2}, T_{ei_2j_2k_2}] \tag{6}$$

Match (i_1, j_1, k_1) and (i_2, j_2, k_2) involve some common athletes, so Match (i_1, j_1, k_1) must be repulsive to match (i_2, j_2, k_2).

$$O[(i, j, k_1), m_1, t_1] = 1 \wedge O[(i, j, k_2), m_2, t_2] = 1 \wedge |t_1 - t_2| \le \tau \Rightarrow m_1 = m_2 \tag{7}$$

Match (i, j, k_1) and (i, j, k_2) involve some common athletes. If they are time-close, they must occupy the same field.

4) Time window constraints

$$[T_{bij_1k}, T_{eij_1k}] \subset [t_1, +\infty] \tag{8}$$

$$[T_{bij_m1}, T_{eij_m1}] \subset [0, t_2] \tag{9}$$

The first round of event i cannot begin earlier than a certain time t_1, and the final match of event i cannot end later than a certain time t_2.

The constraints before describe the Olympics scheduling problem. The target is to gain a solution that satisfies all the constraints. It is a constraint satisfaction problem in fact. As mentioned before, constraint-based methods are back-tracking methods in nature. The efficiency of the constraint-based methods depends on the problem's structure and the back-tracking policy. As the feasible domain gets smaller, the satisfiability would drop abruptly from a certain point, and to find a feasible solution or deciding no solution is very difficult near the point [23]. So we consider model transforming. The feasible domain can be broaden by softening constraints (9), and then we may to try some novel methods.

Let $J = \sum_i w_i T_i^2$ (10)

In (10)

$$T_i = \begin{cases} T_{eij_m1} - T_{ei} & T_{eij_m1} > T_{ei} \\ 0 & else \end{cases} \tag{11}$$

The constraint satisfaction problem can be transformed into a constrained optimization problem as follows:

$\min J$

$s.t.$ (1), (2), (3), (4), (5), (6), (7) and (8) are satisfied.

Constraint (5) can be rewritten as $\mathbf{g(x)} \le \mathbf{0}$.

3 Solution Methodology

3.1 Lagrangian relaxation

Consider the constrained optimization problem. Constraint (1), (2), (3), (4), (6) and (8) are all local constraints, they involve some one kind of field. Constraint (7) does not affect the solution. And constraint (5) is the only global constraints. We relax constraint (5) as follows:

$$L(\mathbf{x},\boldsymbol{\lambda}) = J + \boldsymbol{\lambda}^{\tau}\Big(\sum_{i,j,k,m:S(i)=1} O[(i,j,k),m,t] - \mathbf{m}_{\max}\Big) \tag{12}$$

$\mathbf{x}$ is the vector consists of T_{bijk} and T_{eijk}, $\boldsymbol{\lambda}$ is the vector of Lagrangian multipliers and $\boldsymbol{\lambda} \geq 0$.

The dual problem is

$$q(\boldsymbol{\lambda}) = \min L(\mathbf{x},\boldsymbol{\lambda}) \tag{13}$$

$s.t.$ Constraint (1), (2), (3), (4), (6) and (8) are satisfied.

(12) can be rewritten as follows:

$$L(\mathbf{x},\boldsymbol{\lambda}) = \sum_i L_i - \boldsymbol{\lambda}^{\tau}\mathbf{m}_{\max} \tag{14}$$

$$L_i = w_i T_i^2 + \boldsymbol{\lambda}^{\tau}\Big(\sum_{j,k,m:S(i)=1} \delta[(i,j,k),m,t]\Big] \tag{15}$$

In view of the separability of the original problem, the relaxed problem can be decomposed into I sub-problems as (15).

Due to the duality theorem, $q(\boldsymbol{\lambda}^*) = J^*$. If the optimal Lagrangian multipliers are known, the original problem can be solved by optimizing the separable sub-problems. And if the approximate optimal Lagrangian multipliers are known, the original problem can be approximately solved. We have a Lagrangian relaxation framework as Fig. 1 to solve the constrained optimization problem.

3.2 Lagrangian multipliers updating

The Lagrangian multipliers are the solution of the dual problem, which is proved to be a concave problem. The existing method to solve the dual problem are dependent on variable priori knowledge [18~21]. The subgradient projection method with variable diameter for the dual prolem optimization, which doesnot dependent any priori knownledge, is presented in the following.

It can be concluded that $\mathbf{g}(\mathbf{x})$ is the subgradient of $q(\boldsymbol{\lambda})$ by considering that $q(\boldsymbol{\lambda})$ is concave, and for any $\Delta\boldsymbol{\lambda}$, $q(\boldsymbol{\lambda}+\Delta\boldsymbol{\lambda}) \leq q(\boldsymbol{\lambda}) + \left\langle \mathbf{g}(\mathbf{x}^{\boldsymbol{\lambda}}), \Delta\boldsymbol{\lambda} \right\rangle$.

Let $L(q, q_k^{lev}) = \{\boldsymbol{\lambda} \mid q(\boldsymbol{\lambda}) \geq q_k^{lev}\}$, then $M^* = L(q, q^*)$ is the set of the optimal solutions. Since $q(\boldsymbol{\lambda})$ is difficult to be expressed directly, we rewrite M^* by the subgradient.

$$\bar{q}(\boldsymbol{\lambda};\boldsymbol{\lambda}') = q(\boldsymbol{\lambda}) + \left\langle \mathbf{g}(\mathbf{x}^{\boldsymbol{\lambda}}), \boldsymbol{\lambda} - \boldsymbol{\lambda}' \right\rangle$$

$M^* = \bigcup_{\boldsymbol{\lambda}' \in D} L(\bar{q}(\boldsymbol{\lambda};\boldsymbol{\lambda}'), q^*)$, D is the domain of $\boldsymbol{\lambda}$.

Similarly, the set of the solutions constrained by q_k^{lev} can be rewritten as follows.

$$L(q, q_k^{lev}) = \bigcup_{\lambda' \in D} L(\bar{q}(\boldsymbol{\lambda};\boldsymbol{\lambda}'), q_k^{lev})$$

Then let $q^k(\boldsymbol{\lambda}) = \bar{q}(\boldsymbol{\lambda};\boldsymbol{\lambda}_k)$, $H^k = \{\boldsymbol{\lambda} \mid q^k(\boldsymbol{\lambda}) \ge q_k^{lev}\}$, $\mathbf{y}_k = P_{H^k}(\boldsymbol{\lambda}_k)$. $P_{H^k}(\boldsymbol{\lambda}_k)$ denotes the projection of $\boldsymbol{\lambda}_k$ on H^k. $P_+(\bullet)$ is the operator to project on positive semi-space. $d_S(\bullet)$ denotes the distance to space S. With the closed convex set H^k and an admissible stepsize t, the relaxation operator is $R_{H^k,t}(\boldsymbol{\lambda}) = \boldsymbol{\lambda} + t(P_{H^k}(\boldsymbol{\lambda}) - \boldsymbol{\lambda})$ which has the fejér constraction property

$$\left|\mathbf{y} - R_{H^k,t}(\boldsymbol{\lambda})\right|^2 \le \left|\mathbf{y} - \boldsymbol{\lambda}\right|^2 - t_{\min}(2 - t_{\max}) d_{H^k}^2(\boldsymbol{\lambda}) \qquad \forall\, \mathbf{y} \in H^k .$$

Recall that $H^k = \{\boldsymbol{\lambda} \mid q^k(\boldsymbol{\lambda}) \ge q_k^{lev}\}$, $R_{H^k,t}(\boldsymbol{\lambda}) = \boldsymbol{\lambda} + t(q_k^{lev} - q(\boldsymbol{\lambda}_k))\mathbf{g}(\mathbf{x}_k) / \left|\mathbf{g}(\mathbf{x}_k)\right|^2$.

The method is shown as follows.

Algorithm 1: The subgradient projection method with variable diameter

Step0: Let $k = 0$, $\boldsymbol{\lambda}_0 \ge \mathbf{0}$, ε, δ_0, $d > 0$, $\rho_0 = 0$, $0 < \omega < 1$, $k_d > 1$;

Let $\mathbf{x}_0$ be the solution of $q(\boldsymbol{\lambda}_0) = L(\mathbf{x}_0, \boldsymbol{\lambda}_0)$, and $\mathbf{x}_0^g$ be the corresponding feasible solution；

$J_0 = J(\mathbf{x}_0^g)$, $q_0 = q(\boldsymbol{\lambda}_0)$, $q_0^{lev} = \omega J_0 + (1-\omega) q_0$, $q_0^{up} = J_0$, $\bar{\mathbf{x}} = \mathbf{x}_0^g$;

Step1: If $J_k - q_k \le \varepsilon$, $\bar{\mathbf{x}}$ is the final solution, go to step6; Otherwise, continue;

For $0 < t_{\min} \le t_k \le t_{\max} < 1 - \delta$, δ is a small positive real number

$\boldsymbol{\lambda}_{k+1} = P_+(\boldsymbol{\lambda}_k + t_k(q_k^{lev} - q(\boldsymbol{\lambda}_k)) g(\mathbf{x}_k) / \left|g(\mathbf{x}_k)\right|^2)$;

$\rho_{k+1} = \rho_k + t_k(2 - t_k) d_{H^k}^2(\boldsymbol{\lambda}_k) + d_{R^{N+}}^2(\boldsymbol{\lambda}_k + t_k(P_{H^k}(\boldsymbol{\lambda}_k) - \boldsymbol{\lambda}_k))$;

If $q(\boldsymbol{\lambda}_{k+1}) \ge q_k$, then $q_{k+1} = q(\boldsymbol{\lambda}_{k+1})$; otherwise, $q_{k+1} = q_k$;

$k = k + 1$;

Step2: Let $\mathbf{x}_k$ be the solution of $q(\boldsymbol{\lambda}_k) = L(\mathbf{x}_k, \boldsymbol{\lambda}_k)$, $\mathbf{x}_k^g$ be the corresponding feasible solution;

If $J(\mathbf{x}_k^g) \le J_{k-1}$, then $J_k = J(\mathbf{x}_k^g)$, $\bar{\mathbf{x}} = \mathbf{x}_k^g$; Otherwise, $J_k = J_{k-1}$;

Step3.1: If $\rho_k > d$, then $q_k^{up} = \min\{q_{k-1}^{lev}, J_k\}$, $\rho_k = 0$;

Step3.2: If $\rho_k \le d$, then $q_k^{up} = q_{k-1}^{up}$;

Step4: $0 \le \delta_k \le \delta_{k-1}$, if $q_k > q_k^{lev} - \delta_k$, then $q_k^{up} = J_k$, $d = k_d d$, $\rho_k = 0$;

Step5: $q_k^{lev} = \omega q_k^{up} + (1-\omega) q_k$; go to step1;

Step6: End.

To demonstrate the convergence of the method, we have 3 lemmas and 1 theorem, where $L_f = \sup_{\mathbf{x} \in S} |\mathbf{g}(\mathbf{x})|$, S is the domain of the primal problem's variables.

Lemma 1: if $d < |\boldsymbol{\lambda}_k - \boldsymbol{\lambda}^*|^2$, then there exists some k so that $q_k^{lev} \le q_k + \delta_k$ for δ_k is not small than a certain constant.

Lemma 2: if $d \ge |\boldsymbol{\lambda}_k - \boldsymbol{\lambda}^*|^2$, δ is large enough and $\delta_k = 0$ for any k, then $q_k^{up} \ge q^*$.

Lemma 3: if $d \ge |\boldsymbol{\lambda}_k - \boldsymbol{\lambda}^*|^2$ and $\delta_k = 0$ for any k, then for any $\varepsilon \succ 0$,

$$k > d(L_f)^2 \big/ (t_{\min}(2 - t_{\max})\omega^2 (1-\omega)^2 \varepsilon^2) - ((J(\mathbf{x}_0) - q(\boldsymbol{\lambda}_0))^2 / \varepsilon)^2 \big/ (2e \ln(\omega))$$

$\Rightarrow q_k^{up} - q_k < \varepsilon$.

Theorem 1: if there exists some k^*, so that $d \ge |\boldsymbol{\lambda}_k - \boldsymbol{\lambda}^*|^2$ and $\delta_k = 0$ for $k \ge k^*$, then algorithm1 converges, and the convergence efficiency is

$$k - k^* > d(L_f)^2 \big/ (t_{\min}(2 - t_{\max})\omega^2 (1-\omega)^2 \varepsilon^2) - ((J(\mathbf{x}_0) - q(\boldsymbol{\lambda}_0))^2 / \varepsilon)^2 \big/ (2e \ln(\omega))$$

$\Rightarrow q_k^{up} - q_k < \varepsilon$;

The lemmas and theorem would be proved in the appendix. When some k^* obtained so that $\delta_k = 0$ for $k \ge k^*$, algorithm 1 converges by the efficiency given in theorem 1. The efficiency of the whole algorithm also depends on the efficiency to get k^*, which is determined by the original d and sequence δ_k. If δ_k is large before $d \ge |\boldsymbol{\lambda}_k - \boldsymbol{\lambda}^*|^2$ is obtained and descends to 0 sharply after $d \ge |\boldsymbol{\lambda}_k - \boldsymbol{\lambda}^*|^2$ is obtained, d would gets larger rapidly to exceed $|\boldsymbol{\lambda}_k - \boldsymbol{\lambda}^*|^2$. It is better the original d is more close to $|\boldsymbol{\lambda}_k - \boldsymbol{\lambda}^*|^2$.

Though algorithm 1 converges without any prior knowledge, it is relatively complicated to obtain q_k^{lev}. If the original problem is feasible, we can conclude that $q(\boldsymbol{\lambda}^*) = J^* = 0$, the algorithm could be reduced by fixing $q_k^{lev} = 0$.

3.3 Sub-problems

Each of the sub-problems involves only one event, so it is relatively simple. We solve the sub-problems with enumeration method.

3.4 Construct feasible solutions

Since the field constraints are relaxed, the solutions obtained in the iteration process would violate the field constraints, based on which we must construct feasible solutions. In our work, a heuristic method is developed. The method enumerates all the matches in the order of beginning time, and postpones the current beginning time where the field constraints are violated.

4 Numerical results

The algorithms have been implemented using Matlab and tested on Celeron 4, CPU 1.4 GHz, 256 M SDRAM. Olympics last for 16 days around, but any group of events which are field-coupled one another do not last longer than the track and field, which last for about 10 days. In the tested problems, we deal with the events during 7 days, and each day is partitioned into 3 periods, each period is partitioned into 9 time intervals. Each match lasts for 1~5 intervals, and all matches spread uniformly on the time axis. All the number of intervals and beginning times are generated randomly. The problem involves 20~90 events, for a fixed number of events, the problem is generated and computed 25 times.

The numerical results by algorithm 1 are shown in table 1. $J_{\max}$ is the maximal J that arises in the optimization process, which denotes the maximal cost of violating the final time constraints. And $|q|_{\max}$ is the absolute value of the minimal q, which is the minimal dual value. The difference between J and q is the gap of the original scheduling problem and the dual problem, which is the measurement of the computation error.

If $J_{\min}/J_{\max}$ is smaller than a certain value, it can be decided that a receivable solution is obtained, which violates the final time constraints not badly. If some q>0 is got, it can be decided that the original scheduling problem is unsatisfiable, i.e. there are no solutions which satisfy all the constraints. And as a practical criterion, we can set a positive value $\underline{q}$ to judge that a scheduling problem is unsatisfiable if $q>\underline{q}$.

The ratio of the total time consumed and the number of events is nearly a constant under the fixed steps of iteration. $J_{\min}/J_{\max}$, $|q|_{\min}/|q|_{\max}$, the time to get $J_{\min}$ and the number of tests to get q>0 increase as the number of events increases. All the possible solutions of the problems can be obtained in no more than 10 mins. The phase transition can be recognized by the number of tests to get q>0 and the time first to get q>0. The phase transition is shown in Fig. 2, x-coordinate is the number of events, the y-coordinate are the number of tests not to get q>0 and the average time first to get q>0. From 40 to 70 on x-coordinate, the number of tests having not gotten q>0 decreases fast, and the average time first to get q>0 is distinctly longer than that out of the region.

The numerical results by the reduced algorithm are shown in table 2. The average $J_{\min}/J_{\max}$'s of the variable event number are shown in Fig. 3. And the average time to get $J_{\min}$ by algorithm 1 and the reduced algorithm are shown in Fig. 4. Fig. 3 shows that when the number of events is larger than 50, the average $J_{\min}/J_{\max}$ increases fast. Comparing to the results by algorithm 1, phase transition happens around 50 events. So we can conclude that an original problem is infeasible if $J_{\min}/J_{\max}$ is larger than a certain value. Fig. 4 shows that when the event number is smaller than 40, the difference between the average times to get $J_{\min}$ by algorithm 1 and the reduced algorithm are not notable, when the event number is larger than 40, the time by algorithm 1 is remarkable larger than that by the reduced algorithm, and the difference has a maximal value between 60 and 70. Fig. 3 and 4 cannot provide enough information to recognize the phase transition.

All the above numerical results show that algorithm 1 and the reduced algorithm can both apply to the Olympics scheduling problem. Algorithm 1 is fitter for recognizing the phase transition, and the reduced algorithm is less time consuming.

5 Conclusion

In this paper, the Olympics scheduling problem are modeled as constraint satisfaction problem. By softening the time constraints of the final matches, the constraint satisfaction problem is transformed into a constrained optimization problem. A decomposition methodology based on Lagrangian relaxation is presented for the constrained optimization problem. The dual problem optimization is the key challenge, for which the subgradient projection method with variable diameter are presented. The method is demonstrated to converge, and the efficiency is given. Numerical results show that the methods are efficient, and the phase transition domain can be recognized by algorithm 1.

REFERENCES

[1] George L. Nemhauser, Michael A. Trick, *Scheduling a Major College Basketball Conference*, Operations Research, 46(1): 1-8, 1998.

[2] M. Henz, *Scheduling a Major College Basketball Conference – Revisited*, Operations Research 49 (1): 163-168, 2001.

[3] A. Schaerf, *Scheduling Sport Tournaments Using Constraint Logic Programming*, Constraints, 4(1): 43-65, 1999.

[4] J. C. Regin, *Modeling and Solving Sports League Scheduling with Constraint Programming*, First Congress of the French Operational Research Society (ROADEF), Paris, France, 1998.

[5] K. McAloon, C. Tretkoff, G. Wetzel, *Sports League Scheduling*, Proceeding of the Third ILOG Optimization Suite International Users' Conference, Paris, Franc, 1997.

[6] J. Schönberger, D.C. Mattfeld, H. Kopfer, *Memetic Algorithm timetabling for non-commercial sport leagues*, European Journal of Operational Research, 153 (1): 102–116, 2004.

[7] Yong Tang, *Introduction to temporal database*, Beijing University Publisher, Beijing, China, 2004.

[8] Allen J. F. *Maintaining Knowledge about Temporal Tntervals*, Communications of Association for Computing Machinery, 26(11): 123-154, 1983.

[9] Parthasarathy S. *Building Temporal Constraints into Knowledge Bases for Process Control*, Engineering Applications of Artificial Intelligence, 3(2): 204-209, 1990.

[10] Parthasarathy S. *Building Temporal Constraints into Knowledge Bases for Process Control—Addendum*, Engineering Applications of Artificial Intelligence, 7(1): 83-84, 1994.

[11] Edmund Kieran Burke, Sanja Petrovic, *Recent Research Directions in Automated Timetabling*, European Journal of Operational Research, 140:266-280, 2002.

[12] Peter Luh, Debra J., Hoitomt, Eric Max, KrishnaR. Pattipati, *Schedule Generation and Reconfiguration for Parallel Machines*, IEEE Transactions on Robotics and Automation, 6(6):

687-696, 1990.

[13] Peter Luh, Debra J. Hoitomit, *Scheduling of Manufacturing Systems Using the Lagrangian Relaxation Technique*, IEEE Transactions on Automation Control, 38(7): 1066-1079, 1993.

[14] Hoitomt, D.J.; Luh, P.B.; Pattipati, K.R., *A Practical Approach to Job-shop Scheduling Problems*, IEEE Transactions on Robotics and Automation, 9(1): 1-13, 1993.

[15] Haoxun Chen, Chengbin Chu, Jean-Marie Proth, *An Improvement of the Lagrangean Relaxation Approach for Job Shop Scheduling: A Dynamic Programming Method*, IEEE Transactions on Robotics and Automation, 14(5): 786-795, 1998.

[16] Peter B. Luh, Ling Gou, Yuanhui Zhang, *Job Shop Scheduling with Group-dependent Setups, Finite Buffers, and Long Time Horizon*, Annals of Operations Research, 76: 233 – 259, 1998.

[17] Yuanhui Zhang, Peter B. Luh, Katsumi Narimatsu, Tetsuro Moriya, Tsuyoshi Shimada, Lei Fang, *A Macro-Level Scheduling Method Using Lagrangian Relaxation*, IEEE Transactions on Robotics and Automation, 17(1): 70-79, FEBRUARY 2001.

[18] B. T. Polyak, *Minimization of unsmooth functionals*, USSR Computational Mathematics and Mathematical Physics. 9: 14-29, 1969.

[19] K. C. Kiwiel, *The Efficiency of Subgradient Projection Methods for Convex Optimization, Part I: General Level Methods*. SIAM Journal of Control and Optimization, 34(2): 660-676, 1996.

[20] K. C. Kiwiel, *The Efficiency of Subgradient Projection Methods for Convex Optimization, Part II: Implementation and Optimization*. SIAM Journal of Control and Optimization, 34(2): 677-697, 1996.

[21] S. Kim, H. Ahn, S. Cho, *Variable Target Value Subgradient Method*. Mathematical Programming, 49: 359-369, 1991.

[22] Yongheng Jiang, *Bottleneck Analysis and Coordination of Supply Chain*, Beijing: Ph. D. Dissertion of Tsinghua University, 2003.

[23] P.Cheeseman, B.Kanefsky and w.M.Taylor, *Where the Really Hard Problems Are*, In Proceeding of the 12th international Joint Conference on Artificial Intelligence, 1991.

Appendix

Proof of lemma1:

If there does not exist any k so that $q_k^{lev} \le q_k + \delta_k$, then $q_k < q_k^{lev} - \delta_k$.

Let $q_k^{lev} - q_k - \delta_k = \Delta > 0$, then

$d_{H^k}^2(\boldsymbol{\lambda}_k) = (q_k^{lev} - q(\boldsymbol{\lambda}_k))^2 / |\mathbf{g}(\mathbf{x}_k)|^2 > (\Delta + \delta_k)^2 / (L_f)^2$.

Therefore, there exist a $k^{'}$ large enough so that $\rho_{k'} > d$.

Recall step3.1 and step5, $q_{k'+1}^{lev} - q_{k'+1} = \omega(q_{k'+1}^{up} - q_{k'+1}) \le \omega(q_{k'}^{lev} - q_{k'})$.

By contradiction, there exists some k so that $q_k^{lev} \le q_k + \delta_k$.

The end.♦

Proof of lemma 1:

Without loss of generality, we assume that $\varepsilon = 0$ and the algorithm does not terminate.

Recall that

$\boldsymbol{\lambda}_{k+1} = P_+(\boldsymbol{\lambda}_k + t_k(q_k^{lev} - q(\boldsymbol{\lambda}_k))g(\mathbf{x}_k)/|g(\mathbf{x}_k)|^2) \approx \boldsymbol{\lambda}_k + t_k(q_k^{lev} - q(\boldsymbol{\lambda}_k))\mathbf{g}(\mathbf{x}_k)/|\mathbf{g}(\mathbf{x}_k)|^2$,

And $q(\boldsymbol{\lambda} + \Delta\boldsymbol{\lambda}) \le q(\boldsymbol{\lambda}) + \Delta\boldsymbol{\lambda}^\tau \mathbf{g}(\mathbf{x}^{\boldsymbol{\lambda}})$;

$q(\boldsymbol{\lambda}_{k+1}) - q(\boldsymbol{\lambda}_k) \le t_k(q_k^{lev} - q(\boldsymbol{\lambda}_k))\mathbf{g}^\tau(\mathbf{x}_k)\mathbf{g}^\tau(\mathbf{x}_k)/|\mathbf{g}(\mathbf{x}_k)|^2 + \Delta q$, Δq is a small positive real number, i.e. $q(\boldsymbol{\lambda}_{k+1}) - q(\boldsymbol{\lambda}_k) \le t_k(q_k^{lev} - q(\boldsymbol{\lambda}_k)) + \Delta q$. Notice that δ is large enough, then $q(\boldsymbol{\lambda}_{k+1}) \le q_k^{lev}$.

It can be concluded that step4 never be executed.

We split the iterations into groups $K_0 = \{1 : k(1) - 1\}$ and $K_l = \{k(l) : k(l+1) - 1\}$ if $l \ge 1$. Each group K_l ends by discovering that the target level is unattainable.

By construction, $q_{k+1}^{up} \le q_k^{up}$ and $q_{k+1} \ge q_k$.

Considering step3.1 and step3.2, if $k(l) < k < k(l+1)$, then $q_k^{up} = q_{k-1}^{up}$; therefore, the level $q_k^{lev} = \omega q_k^{up} + (1-\omega)q_k$ cannot decrease: $q_{k+1}^{lev} \ge q_k^{lev}$.

Recall the fejér constraction property, for $\forall \mathbf{y} \in H^{k+1}$

$\rho_{k+1} = \rho_k + t_k(2 - t_k)d_{H^k}^2(\boldsymbol{\lambda}_k) + d_{R^{N+}}^2(\boldsymbol{\lambda}_k + t_k(P_{H^k}(\boldsymbol{\lambda}_k) - \boldsymbol{\lambda}_k)) \le |\mathbf{y} - \boldsymbol{\lambda}_{k(l)}|^2 - |\mathbf{y} - \boldsymbol{\lambda}_{k+1}|^2$.

If $\rho_k > d$, i.e. $\left|\mathbf{y}-\boldsymbol{\lambda}_{k(l)}\right|^2 - \left|\mathbf{y}-\boldsymbol{\lambda}_k\right|^2 > d$, then $\left|\mathbf{y}-\boldsymbol{\lambda}_k\right|^2 < 0$. Therefore, H^k is empty, $q_k^{up} \geq q^*$.

The end. ♦

Proof of lemma3:

For $k(l) < k < k(l+1)$,

$$\rho_k = \sum_{j=k(l)+1}^{k} t_j(2-t_j) d_{H^k}^2(\boldsymbol{\lambda}_j) + \sum_{j=k(l)+1}^{k} d_{R^+}^2(\boldsymbol{\lambda}_j + t_j(P_{H^k}(\boldsymbol{\lambda}_j) - \boldsymbol{\lambda}_j))$$

$$\geq \sum_{j=k(l)+1}^{k} t_j(2-t_j) d_{H^k}^2(\boldsymbol{\lambda}_j)$$

$$\geq \sum_{j=k(l)+1}^{k} t_{\min}(2-t_{\max})(q_k^{lev} - q(\boldsymbol{\lambda}_j))^2 \big/ \left|\mathbf{g}(\mathbf{x}_k)\right|^2$$

For $k(l) < j < k < k(l+1)$ $q_k^{up} - q_k \leq q_j^{up} - q_j$.

Noting that $q_k^{lev} - q(\boldsymbol{\lambda}_k) \geq \omega q_k^{up} + (1-\omega) q_k - q_k = \omega(q_k^{up} - q_k)$ and $\left|\mathbf{g}(\mathbf{x}_k)\right| \leq L_f$, $\rho_k \geq (k-k(l)) t_{\min}(2-t_{\max})(\omega(q_k^{up} - q_k) \big/ L_f)^2$.

Recall that $\rho_k < d$, $k - k(l) \leq d(L_f)^2 \big/ (t_{\min}(2-t_{\max})(\omega(q_k^{up} - q_k))^2)$.

Considering step3 and step5, $q_{k(l)}^{up} - q_{k(l)} \leq (q_{k(l-1)}^{up} - q_{k(l-1)})\omega$

Then $q_{k(l)}^{up} - q_{k(l)} \leq (q_{k(0)}^{up} - q_{k(0)})\omega^l = (J(\mathbf{x}_0) - q(\boldsymbol{\lambda}_0))\omega^l$.

For $\forall \varepsilon > 0$ and $q_{k(m)}^{up} - q_{k(m)} > 0$, $m \leq -(\ln(J(\mathbf{x}_0) - q(\boldsymbol{\lambda}_0)) / \varepsilon) / \ln\omega$.

Notice that $t^2 - 2e\ln t \geq 0$ for $t > 0$, $m \leq -((J(\mathbf{x}_0) - q(\boldsymbol{\lambda}_0)) / \varepsilon)^2 / 2e\ln\omega$.

Recall that $k - k(l) \leq d(L_f)^2 \big/ (t_{\min}(2-t_{\max})(\omega(q_k^{up} - q_k))^2)$, $k(l) < k < k(l+1)$, $k(l) - k(l-1) \leq d(L_f)^2 \big/ (t_{\min}(2-t_{\max})(\omega^{-m+l-1}\varepsilon)^2)$, $0 < l \leq m$

$$k(m) = m + \sum_{l=1}^{m}(k(l) - k(l-1))$$

$$\leq d(L_f)^2 \big/ (t_{\min}(2-t_{\max})\omega^2(1-\omega)^2\varepsilon^2) - (((J(\mathbf{x}_0) - q(\boldsymbol{\lambda}_0))^2 / \varepsilon)^2 / 2e\ln\omega)$$

So $k > d(L_f)^2 \big/ (t_{\min}(2-t_{\max})\omega^2(1-\omega)^2\varepsilon^2) - ((J(\mathbf{x}_0) - q(\boldsymbol{\lambda}_0))^2 / \varepsilon)^2 / (2e\ln(\omega))$

$\Rightarrow q_k^{up} - q_k \prec \varepsilon$.

The end.♦

Proof of theorem1:

By lemma1, if $d < \left|\boldsymbol{\lambda}_k - \boldsymbol{\lambda}^*\right|^2$, some k would be found so that $q_k^{lev} \le q_k + \delta_k$, then step4 would be executed, and d gets larger. So there is some k so that $d \ge \left|\boldsymbol{\lambda}_k - \boldsymbol{\lambda}^*\right|^2$, k^* exists so that $d \ge \left|\boldsymbol{\lambda}_k - \boldsymbol{\lambda}^*\right|^2$ and $\delta_k = 0$ for $k \ge k^*$.

By lemma3, for any $\varepsilon > 0$,

$$k > d(L_f)^2 \big/ (t_{\min}(2 - t_{\max})\omega^2 (1-\omega)^2 \varepsilon^2) - ((J(\mathbf{x}_0) - q(\boldsymbol{\lambda}_0))^2 / \varepsilon)^2 \big/ (2e\ln(\omega))$$

$\Rightarrow q_k^{up} - q_k < \varepsilon$. Where $d \ge \left|\boldsymbol{\lambda}_k - \boldsymbol{\lambda}^*\right|^2$ for $k \ge k^*$

By lemma2, $q_k^{up} \ge q^*$

Considering that the continuity of $q(\boldsymbol{\lambda})$ and the compactness of the domain of $\boldsymbol{\lambda}$, algorithm1 converges.

The end.♦

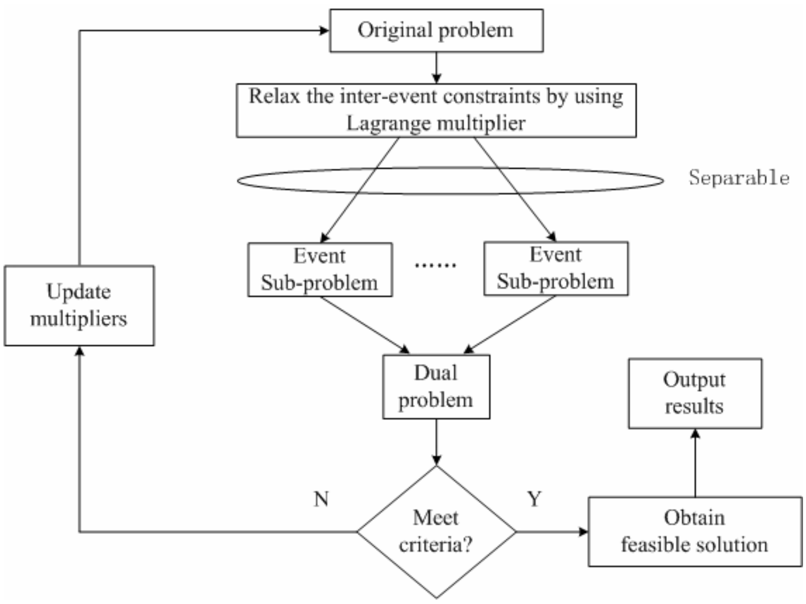

Fig. 1 The Lagrangian relaxation framework

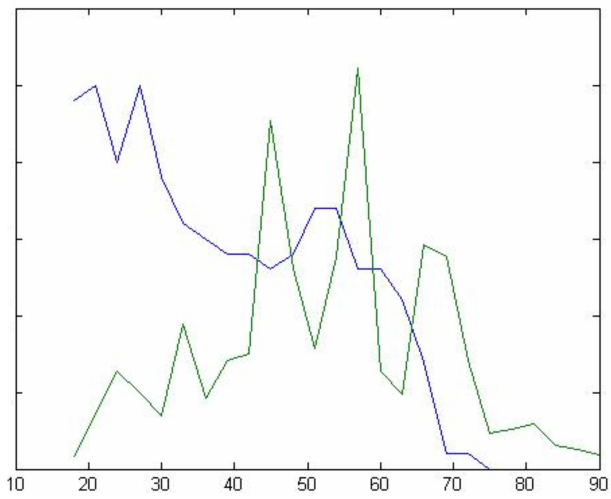

Fig. 2 Phase transition

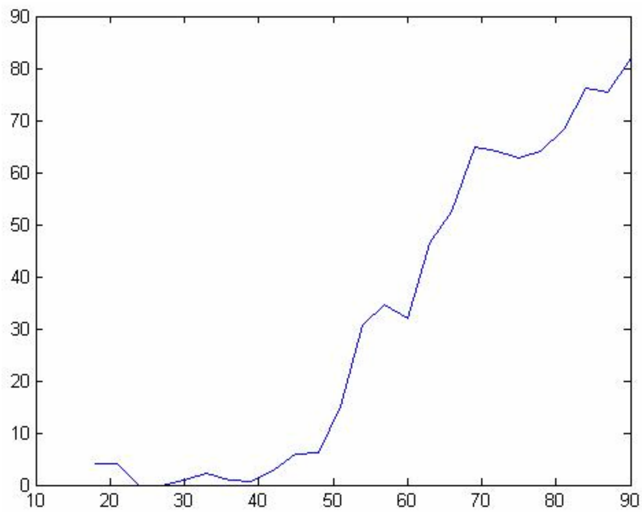

Fig. 3 Average $J_{\min}/J_{\max}$

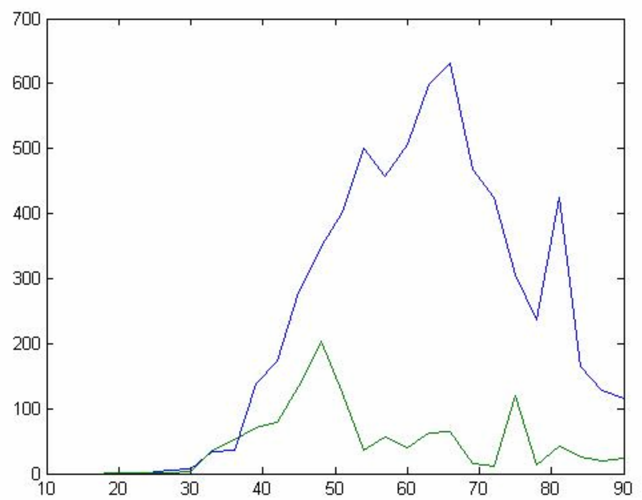

Fig. 4 Average time to get $J_{\min}$

Table 1 The numerical results

Number of events	Steps of iteration	Average time consumed(s)	Average J_{min}/J_{max}	Average time to get J_{min} (s)	Average $\|q\|_{min}/\|q\|_{max}$	Number of tests to get q>0	Average time first to get q>0
18	3000	289.7	0	0.2	0	1	8.0
21	3000	310.9	0	1.4	0	0	-
24	3000	369.1	0	1.2	0	5	64.0
27	3000	499.9	0	4.8	0	0	-
30	3000	523.7	0	7.5	0	6	34.3
33	3000	558.7	0	33.9	0	9	94.2
36	3000	637.6	0.2%	36.8	0.1%	10	45.9
39	3000	662.3	0.9%	138.7	0.8%	11	70.6
42	3000	803.6	1.3%	173.2	0.4	11	75.3
45	3000	822.1	1.7%	278.8	1.6%	12	227.0
48	3000	894.7	3.8%	350.0	4.5%	11	130.4
51	3000	937.6	4.0%	401.5	4.6%	8	78.1
54	3000	1035.3	6.1%	500.4	6.2%	8	138.5
57	3000	1103.7	6.8%	456.5	6.1%	12	261.2
60	3000	1048.9	9.1%	506.0	6.9%	12	63.4
63	3000	1263.0	11.6%	599.3	10.0%	14	49.1
66	3000	1293.7	16.4%	630.9	7.3%	18	146.1
69	3000	1416.9	20.1%	468.2	20.8%	24	137.8
72	3000	1472.7	15.4%	424.2	2.7%	24	70.9
75	3000	1612.5	-	306.8	-	25	23.5
78	3000	1776.0	-	236.0	-	25	25.9
81	3000	1813.6	-	424.2	-	25	29.2
84	3000	1934.8	-	164.2	-	25	15.3
87	3000	2056.8	-	128.0	-	25	12.9
90	3000	2009.2	-	115.9	-	25	9.5

Table 2 The numerical results with q^{lev}=0

Number of events	Steps of iteration	Average time consumed(s)	Average J_{min}/J_{max}	Average time to get J_{min} (s)
18	3000	379.8	4%	0.3
21	3000	419.0	4%	0.6
24	3000	491.9	0	0.5
27	3000	525.7	0	1.8
30	3000	591.9	0.8%	2.4
33	3000	680.0	2.2	35.5
36	3000	732.0	1%	53.1
39	3000	826.1	0.7%	69.8
42	3000	892.7	2.8%	79.3
45	3000	954.6	5.9%	133.3
48	3000	1034.9	6.2%	204.4
51	3000	1137.1	15.1%	123.5
54	3000	1202.1	30.7%	36.7
57	3000	1272.8	34.6%	56.1
60	3000	1373.3	31.9%	39.4
63	3000	1467.3	46.5%	61.5
66	3000	1488.8	52.6%	63.6
69	3000	1510.0	64.8%	14.5
72	3000	1636.3	64%	10.5
75	3000	1689.7	62.8%	120.0
78	3000	1572.3	64.1%	12.7
81	3000	1720.2	68.2%	41.2
84	3000	1807.8	76.2%	24.8
87	3000	1957.9	75.3%	20.4
90	3000	2036.7	82.0%	22.6