

บรรณานุกรม

- ธนัญชัย ลีภักดีปรีดา. (2543). การหาค่าเหมาะสมที่สุด หลักการพื้นฐานและขั้นตอนวิธีการ. กรุงเทพฯ: มหาวิทยาลัยธรรมศาสตร์.
- นันทิ อินทะสอน, เทวัญ เริ่มสูงเนิน. (2553). การจัดตารางการแข่งขันกีฬาโดยใช้กำหนดการจำนวนเต็ม และระเบียบวิธีมอนติ คาร์โล. ขอนแก่น: ภาควิชาคณิตศาสตร์ สาขาวิชาคณิตศาสตร์ประยุกต์ คณะวิทยาศาสตร์ มหาวิทยาลัยขอนแก่น.
- Dirk Briskorn, Andreas Drexl. (2009). IP models for round robin tournaments. **Computers & Operations Research**, 36, 837-852.
- Dirk Briskorn, Sigrid Knust. (2010). Constructing fair sports league schedules with regard to strength groups. **Discrete Applied Mathematics**, 158, 123-135.
- Kelly Easton, George Nemhauser, Michael Trick. (2001). The Traveling Tournament Problem Description and Benchmarks. **Principle and Practices of Constraint Programming, Lecture Notes in Computer Science**, 2239, 580-585.
- K.K.H. Cheung. (2008). Solving mirrored traveling tournament problem benchmark instances with eight teams. **Discrete Optimization**, 5, 138-143.
- Gerhard Post, Gerhard J. Woeginger. (2006). Sports tournaments, home-away assignments and the break minimization problem. **Discrete Optimization**, 3, 165-173.

ภาคผนวก

ภาคผนวก ก

รูปแบบการจัดการแข่งขันสำหรับการแข่งขันที่มีทีมเข้าร่วม 18 ทีม

รูปแบบการจัดการแข่งขันสำหรับการแข่งขันที่มีทีมเข้าร่วม 18 ทีม ที่ปรับปรุงและนำมาใช้ในงานวิจัย

$F1 = [1\ 10\ 13\ 4\ 7\ 16; 2\ 11\ 14\ 5\ 8\ 17; 3\ 12\ 15\ 6\ 9\ 18];$

$F2 = [1\ 8\ 15\ 4\ 11\ 18; 2\ 9\ 16\ 5\ 12\ 13; 3\ 10\ 17\ 6\ 7\ 14];$

$F3 = [1\ 7\ 18\ 2\ 12\ 17\ 5\ 10\ 16\ 6\ 8\ 14\ 4\ 9\ 13\ 3\ 11\ 15];$

$F4 = [1\ 11\ 13\ 5\ 9\ 14; 2\ 12\ 18\ 6\ 10\ 15; 4\ 7\ 17\ 3\ 8\ 16];$

$F5 = [1\ 12\ 14\ 6\ 11\ 16\ 2\ 10\ 18\ 5\ 7\ 15\ 3\ 9\ 17\ 4\ 8\ 13];$

$F6 = [1\ 9\ 15\ 5\ 11\ 17; 2\ 8\ 18\ 4\ 10\ 14; 3\ 7\ 13\ 6\ 12\ 16];$

ภาคผนวก ข

รหัสโปรแกรม

Coding 1 โปรแกรมหลักการประมวลผลด้วยระเบียบวิธี MCG

```

clc;

clear;

lines(0);

MinBP = 900;

sumBP = 0;

MaxBP = 0;

MinT = 100000;

MaxT = 0;

sumT = 0;

r = 10000

for loop = 1 : 10

// phase 1

V = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18];

V = [sufV(V(1,1:6)) sufV(V(1,7:12)) sufV(V(1,13:18))]; // random V in group

F = sufV([1 2 3 4 5 6]); // random pattle

Pin = [inPeriod(6); inPeriod(6) + ones(6,5) * 6; inPeriod(6) + ones(6,5) * 12];

// create pattle in period

X = complete1(V,F,Pin); // complete matching not home-away

// Phase2

xlist = listX(X);

[Xha,Xstat] = home_away(xlist(1:9*17,:),18);

Xstat;

Xstat = showXstat(xlist,Xha,18);

bp = countBreakPoint(Xstat)

```

```

// loop 100

minbp = bp;
minXha = Xha;
minXstat = Xstat;

loop
tic();

for ci = 1 : r

    xlist = listX(X);

    [Xha,Xstat] = home_away(xlist(1:9*17,:),18);

    Xstat = showXstat(xlist,Xha,18);

    bp = countBreakPoint(Xstat);

    if bp < minbp

        minXha = Xha;

        minXstat = Xstat;

        minbp = bp;

    end

end

tt = toc()

minbp

V = minXstat';

if MinBP > minbp

    MinX = X;

    MinBP = minbp;

    MinXstat = minXstat;

end

if MaxBP < minbp

    MaxBP = minbp;

end

sumBP = sumBP + minbp;

sumT = sumT + tt;

```

```
if MinT > tt
```

```
    MinT = tt;
```

```
end
```

```
if MaxT < tt
```

```
    MaxT = tt;
```

```
end
```

```
disp('=====');
```

```
end
```

```
MinX
```

```
MinXstat
```

```
MinBP = MinBP
```

```
Avg = sumBP/30
```

```
MaxBP = MaxBP
```

```
MinT = MinT
```

```
AvgT = sumT/30
```

```
MaxT = MaxT
```

Coding 2 โปรแกรมหลักการประมวลผลด้วยระเบียบวิธี MCG

```

clc;

clear;

lines(0);

// phase 1

MinBP = 900;

sumBP = 0;

MaxBP = 0;

MinT = 100000;

MaxT = 0;

sumT = 0;

r = 1000

rmonti = 10000

nn = 30

for loop = 1 : nn

    loop

    V = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18];

    V = [sufV(V(1,1:6)) sufV(V(1,7:12)) sufV(V(1,13:18))]; // random V in group

    F = sufV([1 2 3 4 5 6]); // random pattle

    Pin = [inPeriod(6); inPeriod(6) + ones(6,5) * 6; inPeriod(6) + ones(6,5) * 12];

    // create pattle in period

    X = complete1(V,F,Pin); // complete matching not home-away

    // Phase2

    xlist = listX(X);

    [Xha,Xstat] = home_away(xlist(1:9*17,:),18);

    Xstat;

    Xstat = showXstat(xlist,Xha,18);

    bp = countBreakPoint(Xstat)

```

```

// init first f0

minbp = bp;

minXha = Xha;

minXstat = Xstat;

minxlist = xlist;

tic();

for ci = 1 : r

    xlist = listX(X);

    [Xha,Xstat] = home_away(xlist,18);

    Xstat = showXstat(xlist,Xha,18);

    bp = countBreakPoint(Xstat);

    if bp < minbp

        minXha = Xha;

        minXstat = Xstat;

        minbp = bp;

        minxlist = xlist;

    end

end

minbpCG = minbp

/// using monti carlo

mc = 20;

mr = 20;

for ci = 1 : rmonti

    xlist = sufxlist(minxlist,mc,mr,18);

    [Xha,Xstat] = home_away(xlist,18);

    Xstat = showXstat(xlist,Xha,18);

    bp = countBreakPoint(Xstat);

    if bp < minbp

        minXha = Xha;

        minXstat = Xstat;

```

```

    minbp = bp;
    minxlist = xlist;
end
end
tt = toc()
minbpMCG = minbp
if MinBP > minbp
    MinX = X;
    MinBP = minbp;
    MinXstat = minXstat;
end
if MaxBP < minbp
    MaxBP = minbp;
end
sumBP = sumBP + minbp;
sumT = sumT + tt;
if MinT > tt
    MinT = tt;
end
if MaxT < tt
    MaxT = tt;
end
disp('=====');
end
r
rmonti
nn
MinX
MinXstat

```

$$\text{MinBP} = \text{MinBP}$$

$$\text{Avg} = \text{sumBP}/\text{nn}$$

$$\text{MaxBP} = \text{MaxBP}$$

$$\text{MinT} = \text{MinT}$$

$$\text{AvgT} = \text{sumT}/\text{nn}$$

$$\text{MaxT} = \text{MaxT}$$

Coding 3 function ที่เรียกใช้

```
function X = alignMatch(V,F)

sV = size(V);

subX = getPattle(F,V);

xx = [];

for j = 1 : sV(2)/2

    xx(1,subX(1,j*2 - 1)) = subX(1,j*2);

    xx(1,subX(1,j*2))    = subX(1,j*2 - 1);

    xx(2,subX(2,j*2 - 1)) = subX(2,j*2);

    xx(2,subX(2,j*2))    = subX(2,j*2 - 1);

end

X = xx;

endfunction
```

```
function [X] = complete1(V,F,Pin)

X = [];

sF = size(F);

sV = size(V);

for i = 1 : sF(2)

    subX = alignMatch(V, F(i));

    X = [X subX'];

    if i ~= sF(2)

        X = [X Pin(:,i)]

    end

end

endfunction
```

```
function [X]=getPattle(T,V)
```

```

F1 = [1 10 13 4 7 16; 2 11 14 5 8 17; 3 12 15 6 9 18];
F2 = [1 8 15 4 11 18; 2 9 16 5 12 13; 3 10 17 6 7 14];
F3 = [1 7 18 2 12 17 5 10 16 6 8 14 4 9 13 3 11 15];
F4 = [1 11 13 5 9 14; 2 12 18 6 10 15; 4 7 17 3 8 16];
F5 = [1 12 14 6 11 16 2 10 18 5 7 15 3 9 17 4 8 13];
F6 = [1 9 15 5 11 17; 2 8 18 4 10 14; 3 7 13 6 12 16];
X = [];
R1 = [];
R2 = [];
if(T == 1) F = F1; end
if(T == 2) F = F2; end
if(T == 3) F = F3; end
if(T == 4) F = F4; end
if(T == 5) F = F5; end
if(T == 6) F = F6; end
sF = size(F);
for i = 1 : sF(1)
    x1 = [];
    for j = 1 : sF(2)/2
        x1 = [x1 V(F(i,2*j -1)) V(F(i,2*j))];
    end
    x2 = [];
    x2 = [x2 V(F(i,sF(2))) V(F(i,1))];
    for j = 1 : sF(2)/2 - 1
        x2 = [x2 V(F(i,2*j)) V(F(i,2*j +1))];
    end
    k = rand();
    if(k < 0.5)
        R1 = [R1 x1];
        R2 = [R2 x2];
    end
end

```

```

else
    R1 = [R1 x2];
    R2 = [R2 x1];
end
end
X = [R1; R2];
endfunction

function [Xha,Xstat] = home_away(Xlist,n)
    sx = size(Xlist);
    X = [Xlist zeros(sx(1),2)]
    Xstat = zeros(1,n);
    Xha = [];
    xha = [];
    i = 1;
    while i <= sx(1)
        if (Xstat(1,Xlist(i,1)) == 2 ) // team1 2 home break
            if(Xstat(1,Xlist(i, 2)) == 2 ) //team2 2 home break too
                // rollback
                [Xstat,xha,i] = rollback(Xstat,Xlist,xha,Xlist(i,1),Xlist(i,2));
            else
                // set team1 away : team 2 home
                xha(i) = 2;
                Xstat = setXstat(Xstat,Xlist(i,2),Xlist(i,1));
            end
        elseif (Xstat(1,Xlist(i,1)) == -2) //team1 2 away break
            if(Xstat(1,Xlist(i,2)) == - 2) //team2 2 away break too
                // rollback
                [Xstat,xha,i] = rollback(Xstat,Xlist,xha,Xlist(i,1),Xlist(i,2));
            else
                // set team1 home : team 2 away
                xha(i) = 1;
            end
        end
        i = i + 1;
    end
end

```

```

        Xstat = setXstat(Xstat,Xlist(i,1),Xlist(i,2));
    end
//change alg
elseif (Xstat(1,Xlist(i,2)) == 2) // team 1 not break but team2 2 home break
    xha(i) = 1;                // set team1 home : team 2 away
    Xstat = setXstat(Xstat,Xlist(i,1),Xlist(i,2));
elseif (Xstat(1,Xlist(i,2)) == -2) // team 1 not break but team2 2 away break
    xha(i) = 2;                // set team1 away : team 2 home
    Xstat = setXstat(Xstat,Xlist(i,2),Xlist(i,1));
else // team1&2 not break
    if Xstat(1,Xlist(i,1)) <= Xstat(1,Xlist(i,2))
        xha(i) = 1;
        Xstat = setXstat(Xstat,Xlist(i,1),Xlist(i,2));
    else
        xha(i) = 2;
        Xstat = setXstat(Xstat,Xlist(i,2),Xlist(i,1));
    end
end
end
i = i + 1;
end
Xha = xha;
endfunction

```

```

function sXstat = setXstat(Xstat,h,a)
    sizexstat = size(Xstat);
    if Xstat(sizexstat(1),h) < 0
        Xstat(sizexstat(1),h) = 1;
    else
        Xstat(sizexstat(1),h) = Xstat(sizexstat(1),h) + 1;
    end
end

```

```
if Xstat(sizeXstat(1),a) > 0;
```

```
    Xstat(sizeXstat(1),a) = -1;
```

```
else
```

```
    Xstat(sizeXstat(1),a) = Xstat(sizeXstat(1),a) - 1;
```

```
end
```

```
sXstat = Xstat;
```

```
endfunction
```

```
function [Xstati,xhai,iii] = rollback(Xstat,Xlist,xha,T1,T2)
```

```
    sxha = size(xha);
```

```
    ii = 1;
```

```
    xha(sxha(1)) = [];
```

```
    for i = 1 : sxha(1) - 1
```

```
        if ((Xlist(sxha(1)-i,1) == T1 | Xlist(sxha(1)-i,2) == T1 | Xlist(sxha(1)-i,1) == T2 | Xlist(sxha(1)-i,2) == T2) &
```

```
        xha(sxha(1)-i) ~= 2)
```

```
            xha(sxha(1)-i) = 2;
```

```
            if max(abs(countStat(Xlist,xha))) < 3
```

```
                ii = sxha(1)-i;
```

```
                break;
```

```
            else
```

```
                xha(sxha(1)-i) = [];
```

```
            end
```

```
        else
```

```
            xha(sxha(1)-i) = [];
```

```
        end
```

```
    end
```

```
    //count new stat
```

```
    Xstat = countStat(Xlist,xha)
```

```
    Xstati = Xstat;
```

```
    xhai = xha;
```

```

    iii = ii;
endfunction

function Xstat = countStat(Xlist,xha) // count status in 1 row
    Xstat = zeros(1,n);
    sxha = size(xha);
    for i = 1 : sxha(1)
        if xha(i) == 1
            Xstat = setXstat(Xstat,Xlist(i,1),Xlist(i,2));
        else
            Xstat = setXstat(Xstat,Xlist(i,2),Xlist(i,1));
        end
    end
endfunction

function X = inPeriod(n)
    subV = [];
    for i = 1 : n
        subV = [subV i];
    end
    X = zeros(n,n-1);
    while sum(X) ~= 105
        X = zeros(n,n-1);
        for i = 1 : n
            sv = subV;
            sv(i) = [];
            sv = sufV(sv);
            for k = 1 : n -1
                for j = 1 : n -1
                    if X(i,j) == sv(k)
                        break;

```

```

end
if j == n-1
    for l = 1 : n-1
        if X(i,l) == 1 & X(sv(k),l) == 1
            X(i,l) = sv(k)
            X(sv(k),l) = i
            break;
        end
    end
end
end
end
end
end
endfunction

```

```

function [Xlist] = listX(X) // sorting x to list
    sx = size(X);
    Xlist = [];
    for j = 1 : sx(2)
        for i = 1 : sx(1)
            if X(i,j) > X(X(i,j),j)
                if(rand() > 0.5)
                    Xlist = [Xlist ; [X(i,j) X(X(i,j),j)]];
                else
                    Xlist = [Xlist ; [X(X(i,j),j) X(i,j)]];
                end
            end
        end
    end
end
end
end

```

endfunction

```
function [X] = setPeriod(V,F,Pin)
```

```
    X = [];
```

```
    sF = size(F);
```

```
    sV = size(V);
```

```
    for i = 1 : sF(2)
```

```
        subX = alignMatch(F(i),V);
```

```
        X = [X subX'];
```

```
    if i ~= sF(2)
```

```
        X = [X Pin(:,i)]
```

```
    end
```

```
end
```

endfunction

```
function Xstat = showXstat(Xlist,Xha,n)
```

```
    // create new Xstat for show
```

```
    Xstat = zeros(1,n);
```

```
    sx = size(Xha);
```

```
    k = 1;
```

```
    countn = 1;
```

```
    for i = 1 : sx(1)
```

```
        if countn > n/2
```

```
            k = k+1;
```

```
            countn = 1;
```

```
            Xstat = [Xstat ; Xstat(k-1,:)];
```

```
        end
```

```
    if Xha(i) == 1
```

```
        Xstat = setXstat(Xstat,Xlist(i,1),Xlist(i,2));
```

```

else
    Xstat = setXstat(Xstat,Xlist(i,2),Xlist(i,2));
end

countn = countn + 1;

end

endfunction

```

```

function [VS] = sufV(V)

VS = [];

sV = size(V);

c = sV(2);

for i = 1 : c

    p = int32(rand()*sV(2)+1);

    VS = [VS V(p)];

    V(p) = [];

    sV = size(V);

end

endfunction

```

```

function Xlist = sufXlist(xlist,c,r,n)

sx = size(xlist);

for i = 1 : c

    k = int32(rand()*sx(1)+1);

    xtmp = xlist(k,1);

    xlist(k,1) = xlist(k,2);

    xlist(k,2) = xtmp;

end

for i = 1 : r

    k = int32(rand()*sx(1)+1);

    kk = k;

```

```
while k == kk
    if modulo(k,n/2) == 0
        kk =( k - n/2) + int32(rand()*n/2 + 1);
    else
        kk =( k - modulo(k,n/2)) + int32(rand()*n/2 + 1);
    end
end
xltmp = xlist(k,:);
xlist(k,:) = xlist(kk,:);
xlist(kk,:) = xltmp;
end
Xlist = xlist;
endfunction
```

เผยแพร่ผลงานวิทยานิพนธ์

ธีระเดช ชนะภาว,เทวัญ เริ่มสูงเนิน.(2554). การจัดตารางการแข่งขันอย่างเป็นธรรมสำหรับฟุตบอลลีกใน
ประเทศไทย. Proceedings of the 16th Annual Meeting in Mathematics (AMM2011); 2011 Mar 10 - 11; Khon
Kaen University, Khon Kaen, Thailand; p 291 - 298

ประวัติผู้เขียน

นายธีระเดช ธนะภา เกิดเมื่อวันที่ 3 พฤษภาคม พ.ศ. 2525 สำเร็จการศึกษาระดับปริญญาตรีวิศวกรรม
ศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น เมื่อปีการศึกษา 2551
เข้าศึกษาในระดับบัณฑิตศึกษา สาขาคณิตศาสตร์ประยุกต์ คณะวิทยาศาสตร์ มหาวิทยาลัยขอนแก่น เมื่อปี
การศึกษา 2551



