บรรณานุกรม

เกศินี วิฑูรชาติ, ธเนตร นรภูมิพิภัชน์, ศรีสมรัก อินทุจันทร์ยง, & เอกรินทร์ ยลระบิล. (2550). **การวิเคราะห์เชิงปริมาณทางธุรกิจ.**พิมพ์ครั้งที่ 7. กรุงเทพ ฯ: พิมพ์ดี.

ชัชวาลย์ เรื่องประพันธ์. (2543). **สถิติพื้นฐาน.** พิมพ์ครั้งที่ 5. ขอนแก่น: ขอนแก่นการพิมพ์.

ผู้บริหาร การตัดสินใจ และการแก้ไขปัญหา. (2552). ค้นเมื่อ 13 เมษายน, จาก http://irrgation.rid.go.th/rid15 /ppn/Knowledge/Decision%20Support%2 0Systems/dss6.htm.

ธนัญชัย ลีภัทดิ์ปรีดา. (2543). **การหาค่าเหมาะสมที่สุด หลักการพื้นฐานและขั้นตอนวิธีการ.** กรุงเทพฯ: มหาวิทยาลัยธรรมศาสตร์.

สุทธิมา ชำนาญเวช. (2552). **การวิเคราะห์เชิงปริมาณ.**พิมพ์ครั้งที่ 4. กรุงเทพฯ: วิทยาพัฒน์.

อุไรวรรณ แย้มนิยม. (2537). **การวิเคราะห์เชิงปริมาณทางการตลาด.** กรุงเทพ ฯ: ภาควิชาการตลาด คณะบริหารธุรกิจ มหาวิทยาลัยรามคำแหง.

A. Anagnostopoulos, L. Michel, P. Van Hentenryck, and Y. Vergados. (2003). A Simulated Annealing Approach to the Traveling Tournament Problem, **CP-AI-OR'2003**.

A. Anagnostopoulos, L. Michel, P. Van Hentenryck, and Y. Vergados. (2006). A Simulated Annealing Approach to the Traveling Tournament Problem**. Journal of Scheduling 9**, 177-193.

Applegate, D. R. Bixby, V. Chvatal, and W. Cook. (1998). On the solution of traveling salesman problems. **Documenta Mathematica Journal der Deutschen Mathematiker Verinigung International Congress of Mathemeticians**, 645-656.

A. Lim, B. Rodrigues, and X. Zhang .(2006). A simulated annealing and hill-climbing algorithm for the traveling tournament problem. **European Journal of Operational Research**, 174(3), 1459-1478.

Boston, Kevin , Bettinger. (1999). Pete An Analysis of Monte Carlo Integer Programming, Simulated Annealing, and Tabu Search Heuristics for Solving Spatial Harvest Scheduling Problems, **Forest Science**, 10, 292-301.

Celso C. Ribeiro, Sebastan Urrutia. (2007). Heuristics for the mirrored traveling tournament problem, **European Journal of Operational Research**, 179, 775–787.

D.T. Connelly. General Purpose. (1992). Simulated Annealing. **European Journal of Operations Research**, 43.

David C. Uthus, Patricia J. Riddle, Hans W. (2009). Guesgen, An ant colony optimization approach to the traveling tournament problem, **Proceedings of the 11th Annual conference on Genetic and evolutionary computation**, July 08-12, 81-83.

Easton, K., G.L. Nemhauser, and M.A. Trick. (2001). The Traveling Tournament Problem: Description and Benchmarks. **Springer Lecture Notes in Computer Science**, 2239, 580-585.

H. Crauwels, D. Van Oudheusden. (2002). A Generate-and-Test Heuristic Inspired by Ant Colony Optimization for the Traveling Tournament Problem. **PATAT 2002**, 314-315.

K. Easton, G. Nemhauser, and M. Trick. (2003). Solving the travelling tournament problem: A combined integer programming and constraint programming approach. **Springer Lecture Notes in Computer Science**, 2740, 101-109.

Gábor Pataki. (2003). Teaching Integer Programming Formulations Using the Traveling Salesman Problem. **Siam Review**, 45(1), 116-123.

GAIL W. DEPUY. (2001). An Integer Programming Heuristic for Component Allocation in Printed Circuit Card Assembly Systems. **Journal of Heuristics**, 7, 351–369.

G. Kendall. (2006). A Constructive Heuristic for the Travelling Tournament Problem. **PATAT**, 443-447.

Gregory Gutin1, Arash Rafiey. (2004). Where n-cycles in n-partite tournaments are longest Cycles. **Discrete Mathematics**, 289, 163-168.

M. Henz. (2001). Scheduling a major college basketball conference revisited. **Operations Research**, 49, 163-168.

Pascal Van Hentenryck and Yannis Vergados. (2006). Traveling Tournament Scheduling: A Systematic Evaluation of Simulated Annealling. **Springer Lecture Notes in Computer Science**, 3390, 228-243.

P.-C. Chen, G. Kendall, and G. V. Berghe. (2007). An ant based hyper-heuristic for the travelling tournament problem. **IEEE Symposium on Computational Intelligence in Scheduling**, 19-26.

P. Van Hentenryck and Y. Vergados. (2007). Population-based simulated annealing for traveling tournaments. **Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence**, 267-272.

Stefan Irnich. (2010). A new branch-and-price algorithm for the traveling tournament problem. **European Journal of Operational Research**, 204, 218–228.

K. Easton. (2009). **Challenge Traveling Tournament Instances.** July 23, 2009, from http://mat.gsia.cmu.edu/TTP/

ภาคผนวก

ภาคผนวก ก

ระยะทางระหว่างจังหวัดในภาคตะวันออกเฉียงเหนือ

ตารางที่ 6 ระยะทางระหว่างจังหวัดในภาคตะวันออกเฉียงเหนือ (กิโลเมตร)

จังหวัด	ร้อยเอ็ด	บุรีรัมย์	สุรินทร์	ศรีสะเกษ	ขอนแก่น	อุดรธานี	หนองคาย	สกลนคร	นครพนม	มหาสารคาม	นครราชสีมา	เลย	ชัยภูมิ	กาฬสินธุ์	ยโสธร	อุบลราชธานี	มุกดาหาร	อำนาจเจริญ	หนองบัวลำภู
ร้อยเอ็ด	0	146	137	230	113	228	279	175	268	40	235	318	216	47	71	169	162	125	274
บุรีรัมย์	146	0	50	155	200	315	366	321	414	145	151	417	192	184	191	216	302	245	361
สุรินทร์	137	50	0	105	256	373	422	312	350	177	198	462	304	184	135	166	246	189	419
ศรีสะเกษ	230	155	105	0	347	462	513	347	332	274	312	552	417	279	159	61	228	136	508
ขอนแก่น	113	200	256	347	0	115	166	205	298	73	190	206	150	77	184	216	243	238	161
อุดรธานี	228	315	373	462	115	0	51	159	252	188	305	152	266	192	303	401	278	357	46
หนองคาย	279	366	422	513	166	51	0	210	303	239	356	202	317	243	350	448	329	404	97
สกลนคร	175	321	312	347	205	159	210	0	93	172	388	311	349	128	230	286	119	207	205
นครพนม	268	414	350	332	298	252	303	93	0	265	481	404	442	221	215	271	104	192	298
มหาสารคาม	40	145	177	274	73	188	239	172	265	0	216	278	176	44	115	213	210	165	234
นครราชสีมา	235	151	198	312	190	305	356	388	481	216	0	344	119	260	272	370	383	326	351
เลย	318	417	462	552	206	152	202	311	404	278	344	0	225	283	391	489	437	445	106
ชัยภูมิ	216	192	304	417	150	266	317	349	442	176	119	225	0	220	261	359	372	315	174
กาฬสินธุ์	47	184	184	279	77	192	243	128	221	44	260	283	220	0	118	216	166	172	238
ยโสธร	71	191	135	159	184	303	350	230	215	115	272	391	261	118	0	98	111	54	349
อุบลราชธานี	169	216	166	61	216	401	448	286	271	213	370	489	359	118	98	0	167	75	447
มุกดาหาร	162	302	246	228	243	278	329	119	104	210	383	437	372	166	111	167	0	88	320
อำนาจเจริญ	125	245	189	136	238	357	404	207	192	165	326	445	315	172	54	75	88	0	324
หนองบัวลำภู	274	361	419	508	161	46	97	205	298	234	351	106	174	238	349	447	320	324	0

ภาคผนวก ข

รหัสโปรแกรม

Coding01 ทำการสร้างผลเฉลยเริ่มต้น

```
function [m,len] = StartMC2(C,r)
  [scm scn] = size(C)
  Mtmp = PreMC2(scm);
  m = [];
  len = 0;
  for i = 1 : scm - 1
    [MC, Len, Mtmp,MCtmp,Inter]=MC2(Mtmp, C, scm, r);
    m = [m; MCtmp];
    disp(sprintf('Len %d : %d',i,Len));
    len = len + Len;
  end
end

function mt = PreMC2(n)
  mt = [];
  for i = 1 : n
    for j = 1 : n
      if i ~= j
        mt = [mt; i, j];
      end
    end
  end
end

function [MC, Len, Mtmp, MCtmp,Inter]=MC2(mt, C, n, r)
% init and back up data
  mttmp = mt;

  Inter = 0;
  f0 = 99999999;
  X = 0;
  MCT = 0;
  mc = 0;
```

```
% Start Loop

  for ir = 1 : r
    crex = 1;
    while crex == 1
      crey = 1;
      try
        mc1 = zeros(n);
        mc2 = zeros(n);
        mt = mttmp;
        mct = [];

        mtmp1 = mttmp;
        mtmp1t = mttmp;
        for in = 1 : n/2
          mt = mtmp1;
          mtt = mtmp1t;
          [smtm smtn] = size(mt);
          if smtm == 0
            crey = 0;
            break
          end
          xt = floor(rand()*smtm + 1 );
          xi = mt(xt,1);
          xj = mt(xt,2);

          mc1(xi,xj) = 1;
          mct = [mct; xi, xj];

          mtmp1 = [];
          for i = 1 : smtm
            if (mt(i,1) ~= xi && mt(i,2) ~= xj && mt(i,1) ~= xj && mt(i,2) ~= xi)
              mtmp1 = [mtmp1; mt(i,:)];
            end
```

```
    end
    mtmp1t = [];
    [smt1m smt1n] = size(mtt);
    if smt1m == 0
      crey = 0;
      break
    end
    for i = 1 : smt1m
      %if (mtt(i,1) ~= xi && mtt(i,2) ~= xj && ( mtt(i,1) ~= xj || mtt(i,2) ~= xi) )
      if (mtt(i,1) ~= xi && mtt(i,2) ~= xj )
        mtmp1t = [mtmp1t; mtt(i,:)];
      end
    end
end

mtmp2 = mtmp1t;
mtmp2t = mtmp1t;
for in = 1 : n/2
  mt = mtmp2;
  %mtt = mtmp2t;
  [smtm smtn] = size(mt);
  if smtm == 0
    crey = 0;
  end
  xt = floor(rand()*smtm + 1 );
  xi = mt(xt,1);
  xj = mt(xt,2);

  mc2(xi,xj) = 1;
  mct = [mct; xi, xj];


  mtmp2 = [];
```

```
        for i = 1 : smtm
          if (mt(i,1) ~= xi && mt(i,2) ~= xj && mt(i,1) ~= xj && mt(i,2) ~= xi)
            mtmp2 = [mtmp2; mt(i,:)];
          end
        end
        %mtmp2t = [];
        %[smt2m smt2n] = size(mtt);
        %for i = 1 : smt2m
        %   if (mtt(i,1) ~= xi && mtt(i,2) ~= xj)
        %      mtmp2t = [mtmp2t; mtt(i,:)];
        %   end
        %end
      end


    if crey == 1
      crex = 0;
    end
    catch
      crex = 1;
      crey = 1;
    end
end

x = zeros(n);
for i = 1 : n/2
  x(mct(i,1),mct(i,2)) = x(mct(i,1),mct(i,2)) + 1;
end

for i = 1 : n/2
  for j = n/2 + 1 : n
    if mct(i,1) == mct(j,1)
      x(mct(i,2),mct(j,2)) = x(mct(i,2),mct(j,2)) + 1;
    end
```

```
        if mct(i,1) == mct(j,2)
            x(mct(i,2),mct(j,2)) = x(mct(i,2),mct(j,2)) + 1;
        end

        if mct(i,2) == mct(j,1)
            x(mct(i,2),mct(j,2)) = x(mct(i,2),mct(j,2)) + 1;
        end

    end
end

for i = n/2 + 1 : n
    x(mct(i,2),mct(i,1)) = x(mct(i,2),mct(i,1)) + 1;
end

f1 = sum(sum(x.*C));

df = f1 - f0;
if df < 0 || exp(-df/0.5) > rand()
    f0 = f1;
    X = x;
    MCT = mct;
    mc = mc1 + mc2;
    if df < 0
        Inter = Inter + 1;
    end

end

end
```

```
    mt = mttmp;
    [smtm smtn] = size(mt);
    [smctm smctn] = size(mct);
    MT = [];
    for i = 1 : smtm
      k = 0;
      for j = 1 : smctm
        if mt(i,1) == MCT(j,1) &&  mt(i,2) == MCT(j,2)
          k = 1;
          break;
        end
      end

      if k == 0;
        MT = [MT; mt(i,:)];
      end
    end


    Len = f0;
    Mtmp = MT;
    MCtmp = MCT;
    MC = mc;
end
```

Coding 02 เปรียบเทียบคัดเลือกผลเฉลย โดยมอนติ คาร์โล

```
function [M,F0,G] = MC2_3_G(mt,C,r,rcol,rrow)
  % เวลาเริ่ม
  t = fix(clock);
  disp(sprintf('เริ่ม %d-%d-%d    %d:%d:%d',t(3),t(2),t(1),t(4),t(5),t(6)));
  %
  M = 0;
  %F0  = 999999999;
  G = [];
  [n nt] = size(C);
  d = n/2; %จำนวนคู่ในแต่ละวัน

  [smtm smtn] = size(mt);
  % หา ระยะทาง f0 จาก mt
  mttmp = [];
  for i = 1 : smtm/d
    ii = (i-1)*d + 1;
    mttmp2 = [];
    for i1 = ii : ii + d - 1
      mttmp2 = [mttmp2 mt(i1,:)];
    end
    mttmp = [mttmp; mttmp2];
  end

  F0 = Caldist(mttmp,n,C);

  %M = mt;
  disp(sprintf('เริ่มต้น F0 = %d\n',F0));
  ri = 0;
  for i = 1 : r
    mt2 = mt;
        %innt = floor(rand()* (rcol+1));
      %for inn = 1 : innt
```

```
    %if (rand() < 0.5)
  innt = floor(rand()* (rcol+1));
  for inn = 1 : innt
      ft = floor(rand()*smtm)+1;
      xi = mt2(ft,1);
      xj = mt2(ft,2);
      for k = 1 : smtm
        if mt2(k,1) == xj && mt2(k,2) == xi
          mt2tmp = mt2(ft,:);
          mt2(ft,:) = mt2(k,:);
          mt2(k,:) = mt2tmp;
          break;
        end
      end
  end
        innt = floor(rand()* (rrow+1));
  for inn = 1 : innt
    %else
      ft1 = floor(rand() * (n-1) * 2) + 1;
      ft2 = floor(rand() * (n-1) * 2) + 1;
      while ft1 == ft2
        ft2 = floor(rand() * (n-1) * 2) + 1;
      end
      for k = 1 : d
        mt2tmp = mt2((ft1-1)*d + k,:);
        mt2((ft1-1)*d + k,:) = mt2((ft2-1)*d + k,:);
        mt2((ft2-1)*d + k,:) = mt2tmp;
      end
  end
% คำนวณระยะทาง f1
mttmp = [];
for k = 1 : smtm/d
  kk = (k-1)*d + 1;
```

```
        mttmp2 = [];
        for k1 = kk : kk + d - 1
          mttmp2 = [mttmp2 mt2(k1,:)];
        end
        mttmp = [mttmp; mttmp2];
      end
      f1 = Caldist(mttmp,n,C);
       %% = mttmp;
      df = f1 - F0;
      if df < 0 || rand() < exp(-df/0.5)
        if i ~= 1
          G = [G;i-1, F0];
        end
        F0 = f1;
        mt = mt2;
    %%    disp(sprintf('\t\t\t\t\t\tยอมรับ F1 = %d',f1));
        if df < 0
          t = fix(clock);
          ri = i;
        end
         %disp(sprintf('1\t%d\t%d',F0,i));
        G = [G;i-1, F0];
      else
      end
    end
    disp(sprintf('เวลาเจอระยะทางสั้นสุด %d-%d-%d    %d:%d:%d',t(3),t(2),t(1),t(4),t(5),t(6)));
    t = fix(clock);
    disp(sprintf('เสร็จสิ้น %d-%d-%d    %d:%d:%d',t(3),t(2),t(1),t(4),t(5),t(6)));
    disp(sprintf('รอบที่พบเหมาะสมที่สุด : %d',ri));
    M = mt;
    %plot(G(:,1),G(:,2));
End
```

Coding03 การคำนวณ Y และหาค่า F

```
function dis = Caldist(m,n,C) %m matrix n town C length
  [sm sn] = size(m);
  Y = zeros(n,n);
    %ggg = m
  for i = 1 : sm
    for j = 1 : sn/2
      if(mod(i,2) == 1)
        Y(m(i,(j*2)-1),m(i,(j*2))) = Y(m(i,(j*2)-1),m(i,(j*2))) + 1;
      else
        % ตรวจสอบทีมเหย้า
        for jj = 1 : sn/2
          if m(i,(j*2)) == m(i-1,(jj*2)) % เหย้า >> เหย้า
            %kk = 1;
            %break;
          end
          if m(i,(j*2)) == m(i-1,(jj*2)-1) % เยือน >> เหย้า
            %kk = jj;
            Y(m(i-1,(jj*2)),m(i,(j*2))) = Y(m(i-1,(jj*2)),m(i,(j*2))) + 1;
            %break;
          end
          if m(i,(j*2)-1) == m(i-1,(jj*2)) % เหย้า >> เยือน
            Y(m(i,(j*2)-1),m(i,(j*2))) = Y(m(i,(j*2)-1),m(i,(j*2))) + 1;
          end
          if m(i,(j*2)-1) == m(i-1,(jj*2)-1) % เยือน >> เยือน
            Y(m(i-1,(jj*2)),m(i,(j*2))) =  Y(m(i-1,(jj*2)),m(i,(j*2))) + 1;
          end
        end
              Y(m(i,(j*2)),m(i,(j*2)-1)) = Y(m(i,(j*2)),m(i,(j*2)-1)) + 1; % เดินทางกลับ
      end
    end
  end
  dis = sum(sum(Y.*C));
end
```

เผยแพร่ผลงานวิทยานิพนธ์

นิ่ม อินทะสอน ,เทวัญ เริ่มสูงเนิน.(2553). **การจัดตารางการแข่งขันกีฬาโดยใช้กำหนดการจำนวนเต็ม และระเบียบวิธีมอนติ คาร์โล.** การจัดประชุมวิชาการเสนอผลงานวิจัยระดับบัณฑิตศึกษาครั้งที่ 4 บัณฑิตวิทยาลัย มหาวิทยาลัยราชภัฏสุรินทร์ .จังหวัดสุรินทร์ 20-22 กุมภาพันธ์ 2553

ประวัติผู้เขียน

นางสาวนิ่ม อินทะสอน เกิดเมื่อวันที่ 26 เมษายน พ.ศ. 2526 สำเร็จการศึกษาระดับปริญญาตรีวิทยาศาสตรบัณฑิต สาขาคณิตศาสตร์ จากวิทยาศาสตร์ มหาวิทยาลัยขอนแก่น เมื่อปีการศึกษา 2548 เข้าศึกษาในระดับบัณฑิตศึกษา สาขาคณิตศาสตร์ประยุกต์ คณะวิทยาศาสตร์ มหาวิทยาลัยขอนแก่น เมื่อปีการศึกษา 2549