

Algoritma Pembenaman Perlakuan Individu Semut (Agen) ke Dalam Sistem Koloni Semut

ZULAIHA ALI OTHMAN, MOHD HELMI RAIS,
ROSLAN AHMAD & ABDUL RAZAK HAMDAN

ABSTRACT

Ants are fascinating creatures that demonstrate the ability to find food and bring it back to their nest. Their ability as a colony, to find paths to food sources has inspired the development of algorithms known as Ant Colony Systems (ACS). The principle of cooperation forms the backbone of such algorithms, commonly used to find solutions to problems such as the Traveling Salesman Problem (TSP). Ants communicate to each other through chemical substances called pheromones. Modeling individual ants' ability to manipulate this substance can help an ACS find the best solution. This paper introduces a Dynamic Ant Colony System with three-level updates (DACS3) which embeds individual ant behavior(agent) in existing ACS. Experiments were conducted to observe single ant behavior in a colony of Malaysian House Red Ants. Such behaviour was incorporated into the DACS3 algorithm. We benchmarked the performance of DACS3 against several Ant Colony Optimizations (ACO) algorithms on TSP instances ranging from 14 to 100 cities. The result showed that embedding a simple behaviour into an ant improved an ACS algorithm.

Keywords: Ant Colony System, Optimisation, Traveling Sales Problem.

ABSTRAK

Semut adalah suatu hidupan yang terkenal dengan sifat kerjasama antara satu sama lain untuk mendapatkan makanan dan membawa balik makanan tersebut kesarangnya. Kebolehan secara berkoloni terutamanya bagi menghasilkan laluan terbaik untuk mendapatkan makanan telah menjadi sumber inspirasi kepada pembangunan algoritma yang dikenali sebagai Sistem Koloni Semut (SKS). Prinsip kooperasi antara semut adalah tulang belakang kepada pembangunan algoritma ini untuk menghasilkan laluan terdekat dalam Permasalahan Perjalanan Jurujual (PPJ). Phoremone adalah bahan kimia yang dihasilkan oleh semut sebagai perantara untuk mereka berkomunikasi antara satu sama lain. Perlakuan individu semut (Agen) berupaya memanipulasi paras phoremone dan seterusnya membantu SKS menghasilkan penyelesaian terbaik. Tujuan kertas kerja ini adalah untuk memperkenalkan satu algoritma Sistem Koloni Semut Dinamik Pengemaskinian Tiga Paras (SKSD3) yang dibina hasil pembenaman perlakuan individu semut dalam SKS yang ada. Model perlakuan individu semut dicerap daripada hasil beberapa ujikaji terhadap perlakuan seekor semut dalam sistem koloni semut merah Malaysia. Keupayaan SKSD3 diukur dengan membandingkannya dengan beberapa algoritma Koloni Semut set data PPJ julat di antara 14 hingga 100 bandar. Hasil ujikaji menunjukkan bahawa pembenaman perlakuan individual semut dalam SKS telah memperbaiki algorithms tersebut.

Kata kunci: Sistem Koloni Semut, Pengoptimuman, Permasalahan Perjalanan Jurujual.

PENGENALAN

Kebanyakan persekitaran bisnes masa kini secara tidak langsung terlibat dalam industri pengangkutan dan logistik yang semakin kompleks, dinamik serta menyediakan operasi yang fleksibel. Dua cabaran utama syarikat: memenuhi kehendak pelanggan yang sering membuat permintaan yang berubah-ubah, sedangkan operasi penghantaran telah dilakukan dan menghasilkan penyelesaian laluan terbaik bagi menjimatkan kos penghantaran. Keupayaan manusia untuk membuat pemilihan laluan terbaik adalah terhad apabila masalah tersebut adalah besar dan kompleks dari segi saiz dan kekangannya.

SKS adalah satu daripada algoritma yang terkenal digunakan dalam penyelesaian masalah pengotimuman seperti masalah perjalanan jurujual, penskedulan kerja, masalah perjalanan kenderaan, masalah umpukan kuadratik (Dorigo et al. 1999; Dorigo & Caro 1999). Idea algoritma ini dicipta adalah hasil daripada penelitian perlakuan sekumpulan semut yang berkomunikasi melalui satu bahan yang dikenali sebagai phorenone. Phorenone ini beraksi sebagai pembolehkan ingatan yang digunakan bagi membantu semut lain menjana laluan terbaik (Fleisher 2003). SKS terkenal sebagai satu kaedah mendapatkan laluan terdekat perjalanan antara bandar yang dijejaki (Dorigo et al. 1999). Semua bandar yang berhubung antara satu sama lain hanya boleh didatangi sekali sahaja. Bagi mengatasi masalah permintaan pengguna yang berubah-ubah, penskedulan laluan perjalanan dijana semua apabila tempahan diminta dengan mengambilkira proses penghantaran barangan telah berjalan. Objektif utama dinamik PPJ adalah untuk menghasilkan laluan terbaik dengan anggapan tiada kekangan jumlah barangan yang dibawa. Jadi, masa menjana penyelesaian terbaik adalah menjadi faktor penting kerana sistem ini lebih fleksibel dan perubahan tempahan baru diurus dengan lebih cekap. Banyak versi SKS telah dihasilkan bagi menghasilkan laluan terdekat dan menangani isu dinamik ini. Bagaimana pun hasil kajian mendapati SKS yang ada masih mempunyai kekuatan dan kelemahannya yang tersendiri. Penghasilan jarak terdekat dan masa untuk mendapatkan jarak terdekat berserta saiz bandar adalah kayu ukur kepada prestasi sesuatu algoritma sistem koloni itu.

Fokus utama kajian adalah untuk menambahbaikkan algoritma SKS yang ada dengan mengambil idea bahawa seekor semut dalam koloni itu mempunyai perlakuan. Perlakuan semut ini dipercayai berupaya mengubah paras phorenone apabila semut membuat laluan. Adalah dijangkakan bahawa dengan pembedaan perlakuan individu semut dalam sistem koloni dapat menghasilkan jarak terdekat lebih dekat serta menghasilkan masa pelaksanaan yang lebih cepat. Pembedaan perlakuan individu semut dalam SKS ini menghasilkan satu algoritma yang dinamakan Sistem Koloni Semut Dinamik Tiga Paras Kemaskini (SKSD3). SKSD3 dibangunkan berasaskan algoritma SKS yang dibangunkan oleh Morco Dorigo (Dorigo and Maniezzo 2001) tetapi paras kemaskini adalah berdasarkan pembaikan algoritma SKSD. Model asas SKSD adalah satu algoritma yang bersesuaian untuk penyelesaian PPJ dinamik (Li and Gong 2003).

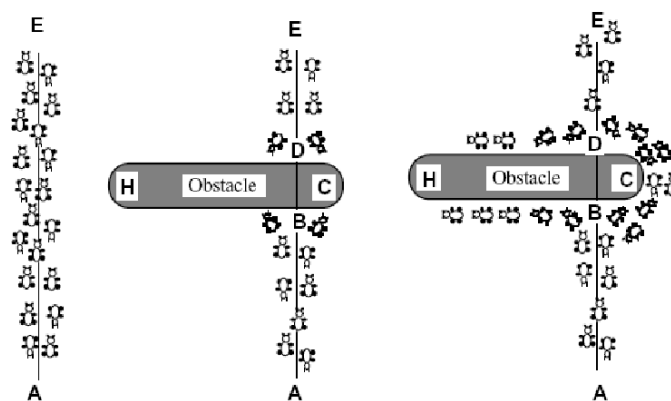
Kertas kerja ini dibahagikan kepada tujuh seksyen. Seksyen kedua membincangkan kaedah ujikaji yang dilakukan untuk mencerap tabiat koloni semut bagi menjanakan algoritma SKS. Kaedah ujikaji untuk mencerap tabiat individual semut serta model perlakuan semut juga dibincangkan. Seksyen tiga pula membincangkan beberapa algoritma sistem koloni semut yang lepas yang berkaitan dengan kajian ini, seterusnya seksyen empat membincangkan model dan algoritma SKSD3 yang menggambarkan bagaimana perlakuan individual semut dibenamkan dalam SKS. Seksyen lima, menjelaskan penyediaan ujikaji yang dijalankan dan dua seksyen terakhir memaparkan dan membincangkan keputusan, serta cadangan kajian akan datang bagi mempertingkatkan prestasi SKSD3.

PERLAKUAN SEMUT

Semut adalah serangga yang bersosial dengan cara berkomunikasi dan berkerjasama antara satu individual semut yang lain dan dengan sekumpulan semut yang lain. Mereka berkomunikasi menggunakan satu media kimia yang dipanggil phorenone yang melekat pada tanah sebagai tanda adanya laluan. Tanda ini juga

digunakan oleh semut sewaktu kembali ke sarangnya. Tanda ini juga digunakan oleh semut yang lain untuk mengenalpasti bahawa laluan tersebut mungkin sebagai laluan terbaik untuk mengangkut sumber makanan ke sarang dan sebaliknya. Jumlah phorenone yang terletak pada laluan menentukan laluan pilihan semut. Kajian sosial semut ini telah dikaji sejak abad 50 lagi, bila French Entomologist iaitu Piera Paul Grass mengkaji bagaimana anai-anai berkomunikasi (Grosan and Abraham 2006). Beliau juga mendapati bahawa sosial serangga bertindak balas dengan adanya rangsangan signal yang diaktifkan dan secara genetiknya mengkodkan reaksi yang dipanggil 'stimergy' (Dorigo & Socha 2006). Komunikasi ini dikelaskan sebagai komunikasi secara langsung yang membolehkan anai-anai teruja. Sifat 'stimergy' juga ada pada semut bila Derembourg et al. menjalankan kajian ke atas status dua laluan terhadap semut Argentina. Hasil menunjukkan bahawa kewujudan phorenone tidak menjejaskan perlakuan semut tersebut (Deneubourg et al. 1990).

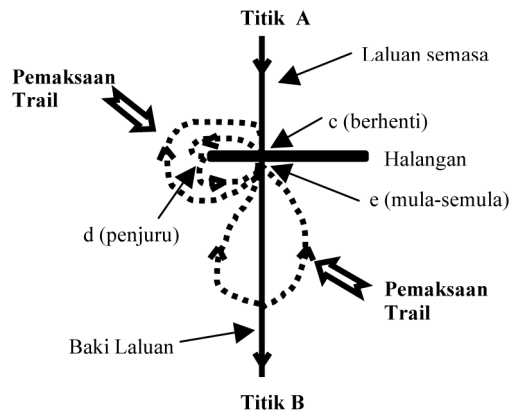
Berdasarkan kajian ini, Dorigo et al. telah menukarkan perlakuan ini kepada semut buatan. Hasil kajiannya menunjukkan bahawa laluan terdekat adalah berdasarkan jumlah paras phorenone terbanyak. Beliau juga mengkaji tentang pergerakan rawak semut yang dikenali sebagai keadaan kebingungan bila halangan diberi. Kajian mendapati semut tersebut berjaya mengenalpasti laluan sebelum ini dan membentuk laluan sendiri. Perlakuan kebingungan ini adalah satu proses pemaksaan pembentuk laluan semula phorenone. Rajah 1 menunjukkan hasil ujikaji yang dilakukan oleh Dorigo secara spontan, bila satu halangan diletakkan di tempat laluan semut yang banyak.



RAJAH 1. Ujikaji Halangan

Prinsip kooperasi antara kumpulan serangga ialah komunikasi antara individu serangga yang menyumbang kepada keseluruhan kumpulan serangga untuk terus hidup (Deneubourg et al. 1990). Beberapa ujikaji dijalankan ke atas koloni semut merah (semut gatal) Malaysia menggunakan kaedah ujikaji Dorigo iaitu sebuah objek atau halangan diletakkan diatas laluan seekor semut untuk mengetahui perlakuan individu semut ini (Rais et al. 2007). Ujikaji ini dijalankan bila mana semut tidak begitu banyak dalam koloninya. Ini untuk mengelakkan gangguan mencerap perlakuan individu semut dan biasanya dijalankan pada lewat malam.

Rajah 2 menunjukkan perlakuan seekor semut membina semula laluan. Semut akan cuba berjalan terus menggunakan laluan sedia ada iaitu Titik A dan Titik B menggunakan petua paras pheromone. Apabila satu penghalang diletakkan pada laluan tersebut, semut mula mencari laluan alternatif. Ini bermula dengan memeriksa kedua-dua penjuru penghalang beberapa kali. Kemudian ia memilih penjuru yang terpendek, meneruskan perjalanannya dan mula membina laluan dengan melawati penjuru terpendek di *d* dari *c*, kemudian ia mencuba untuk mendapatkan titik *e* iaitu titik yang mengembalikannya ke laluan asal. Apabila ini telah dilakukan maka, satu laluan baru dibina dengan melawati *c*, *d*, *e* beberapa kali.



RAJAH 2. Perlakuan Seekor Semut

Apabila satu laluan alternatif telah dibina sepenuhnya, semut tersebut akan meneruskan perjalanan menggunakan laluan sebelum diberi penghalang. Walau bagaimanapun, selepas beberapa jarak, terdapat beberapa semut yang kembali semula ke titik c , iaitu titik mula perjalanan dan terus memaksa untuk melawat semula laluan itu beberapa kali. Daripada penelitian ini, kesimpulan dapat dibuat, untuk membina suatu laluan, seekor semut perlukan tiga even: pertamanya, pembinaan laluan dan dua even lagi digunakan untuk mengukuhkan laluan tersebut.

ALGORITMA PENGOPTIMUMAN KOLONI SEMUT

SISTEM SEMUT

Darigo et al. memperkenalkan dan mengaplikasikan sistem semut dalam PPJ. Pada mulanya, semut diletakkan pada n buah bandar, ia bergerak daripada bandar r ke bandar s menggunakan formula kebarangkalian yang dipanggil petua rambang berkadar S seperti berikut:

$$S = \begin{cases} \frac{[\tau(r,s)\eta(r,s)]^\beta}{\sum_{u \in J_k(r)} [\tau(r,u)\eta(r,u)]^\beta} & \text{if } s \in J_k(r), \\ 0 & \text{Selainnya} \end{cases} \quad (1)$$

Formula jarak Eulidean iaitu $(d_{ij} = ([xi-xj]^2 + [yi-yj]^2)^{1/2})$ digunakan untuk mengira jarak antara bandar r dan bandar s . $\tau(r,s)$ ialah ketumpatan laluan pada penjuru (i,j) dan kewujudan $\eta(r,s) = 1/d_{ij}$ adalah jarak sonsang pada pergerakan (r,s) . β adalah pembolehubah ralat yang digunakan untuk menentukan mana satu yang lebih penting, diantara jarak berbanding paras phoremone ($\beta > 0$). $J_k(r)$ ialah satu set bandar yang perlu dilawati oleh semut k yang terletak dibandar r . Selepas semua semut telah menamatkan lawatannya, paras phoremone pada setiap penjuru perlu dikemaskini menggunakan peraturan kemaskini phoremone tempatan berikut (Dorigo & Caro 1999).

$$\tau(r,s) \leftarrow (1-p)\tau(r,s) + \sum_{k=1}^m \Delta\tau_k(r,s)$$

di mana

$$\Delta\tau_k(r,s) = \begin{cases} 1/L_k & \text{if } (r,s) \in \text{perjalanan oleh semut } k \\ 0 & \text{Selainnya} \end{cases}$$
(2)

$(1-p)$ adalah pembolehkan phorenone yang tertinggal bila ($0 < p < 1$) menunjukkan laluan yang teruap, bila semut memilih suatu bandar untuk bergerak. L_k adalah jarak perjalanan yang dilalui oleh semut k manakala m ialah jumlah semut yang digunakan.

SISTEM KOLONI SEMUT

Gambardella et al. (Anderson & Franks 1999) pula, telah mengubahsuai algoritma koloni semut ini. Ia telah memperkenalkan Sistem Koloni Semut. Sistem ini menyediakan keadaan yang lebih seimbang dan menyediakan bantuan mengelintar dengan 3 kaedah. Pertama menggunakan peraturan keadaan transiti iaitu mengabungkan fomular persamaan (1) dan persamaan (2) yang berupaya memperbaiki arahan bagi mengimbangi antara menelajah laluan baru atau cuba eksplloit bandar yang dijejaki sebelumnya (persamaan (3)). Kedua, setiap penjuru yang dipunyai oleh laluan semut terbaik sahaja dibenarkan untuk kemaskini phorenone menggunakan peraturan kemaskini phorenone global. Phorenone tempatan juga diaplikasikan bila semut mula membentuk suatu penyelesaian (Gaertner & Clark 2005).

$$s = \begin{cases} \arg \max_{u \in J_k(r)} [\tau(r,u) \text{In}(r,u)]^q & \text{if } q \leq q_0 \text{ (Eksplloitasi)} \\ S & \text{Selainnya (Penerokaan)} \end{cases}$$
(3)

Setiap semut yang membina laluan lawatannya berulang kali menggunakan peraturan kebarangkalian keadaan. q adalah nombor rawak yang diagihkan secara sekata $[0..1]$ dan q_0 ialah parameter $0 < q_0 < 1$, yang menentukan pilihan terbaik antara ekploitasi atau penerokaan.

Semasa pembinaan laluan, semut akan mengubah paras kumuhan yang ada pada penjuru yang dilawatnya menggunakan peraturan kemaskini tempatan (persamaan (4)). L_{mn} adalah jarak perjalanan yang dihasilkan menggunakan kaedah heuristik jiran yang terdekat dengan n adalah jumlah bandar yang dijejaki.

$$\tau(r,s) \leftarrow (1-p)\tau(r,s) + p \cdot \Delta\tau_0$$
(4)

di mana $\Delta\tau_0 = (1/L_{mn} \cdot n)$

Setelah semua semut membina laluan, hanya semut yang paling baik secara keseluruhannya, iaitu yang menghasilkan laluan terdekat daripada tempat mula lawatan yang dibenarkan mengemaskini paras kumuhan menggunakan peraturan kemaskini phorenone global (persamaan (5)). L_{gs} ialah jarak terbaik lawatan keseluruhan daripada tempat mula lawatan

$$\tau(r,s) \leftarrow (1-p)\tau(r,s) + p \cdot \Delta\tau(r,s)$$

di mana $\Delta\tau(r,s) = \begin{cases} (L_{gs})^{-1} & \text{if } (r,s) \in \text{Laluan Terbaik Global} \\ 0 & \text{Selainnya} \end{cases}$

(5)

SISTEM SEMUT MAX-MIN (MMAS)

Stutzle dan Hoos (Dorigo & Caro 1999) mempertimbangkan SKS semasa membangunkan MMAS, dalam masa yang sama beliau telah membuat penambahbaikan pada Sistem Semut. MMAS berbeza dengan tiga keadaan. Pertamanya ialah menghadkan hanya seekor semut sahaja yang boleh mengemaskini phoremone. Ini sama dengan konsep SKS, tetapi ia ada pilihan samada boleh kemaskini satu penyelesaian pada lalaran semasa atau mengambil penyelesaian terbaik keseluruhan. Kedua, phoremone dibahagikan kepada 2 julat iaitu had bawah dan atas $[t_{max}, t_{min}]$ untuk mengelakkan gelintaran menjadi tepu. Ketiga, nilai awal paras phoremone disetkan sebagai t_{max} , dengan cara ini membolehkan keadaan gelintaran yang lebih banyak daripada mula algoritma dilaksanakan (Gendreau & Potrin 2005). Algoritma MMAX adalah seperti berikut:

$$\tau(r, s) = p \cdot \tau(r, s) + \Delta \tau_{rs}^{best} \quad (6)$$

di mana $\Delta \tau_{rs}^{best} = 1 / f(s^{best})$

Persamaan (6) menunjukkan bagaimana MMAS mengkemaskini phoremone. S^{best} mengambil nilai samada penyelesaian global daripada mula perjalanan atau penyelesaian pada lalaran semasa. Walaubagaimana pun MMAS, lebih gemar menggunakan penyelesaian lalaran semasa. Dengan cara memberikan pilihan sebegini, penyelesaian yang mempunyai frekuensi tinggi akan menerima jumlah memaksakan phoremone yang lebih besar. Ini akan menghasilkan keputusan yang berbeza dari satu lalaran ke satu lalaran yang lain.

Jadi, satu mekanisma penstabilan diperkenalkan dalam MMAS yang dikenali sebagai Perlembutan Laluan Phoremone (PLP) yang ditunjukkan dalam persamaan (7). PLP ini digunakan untuk mempertingkatkan perlaksanaan. Bila penyelesaian menggunakan MMAS menghampiri penumpuan, mekanisma ini dapat membantu meningkatkan lagi kebarangkalian laluan phoremone ke paras maksimum. Kebaikan mekanisma ini membolehkan maklumat dikumpul semasa perlaksanaan dan tidak hilang sepenuhnya tetapi agak lemah. Mekanisma ini sangat sesuai bagi melaksanakan sistem semut yang lebih panjang.

$$\tau(r, s)^* = \tau(r, s) + \delta \cdot (\tau_{max} - \tau(r, s)) \quad \text{dengan } 0 < \delta < 1$$

di mana $\tau_{max} = \frac{1}{1-p} \cdot \frac{1}{f(S^{bb})}$ (7)

SISTEM KOLONI SEMUT DINAMIK (SKSD)

Sistem Koloni Semut Dinamik adalah sistem yang diperkenalkan bagi menangani masalah PPJ dinamik. Konsep PPJ dinamik menjadikan sistem lebih fleksibel yang mana pengguna boleh mengubah tempahan pada bila-bila masa dan menyediakan penghantaran barangan lebih cepat dan menjimatkan pihak syarikat. PPJ dinamik menggunakan konsep penyusunan semula laluan termasuk kenderaan dalam perjalanan. Kaedah dinamik ini merupakan pembahagian julat masa untuk penskedulan semua laluan terbaik. Bagaimana pun, muatan dianggap bukan sebagai kekangan.

Li dan Gong telah memperkenalkan satu lagi algoritma hasil penambahbaikan sistem semut yang asal dengan menambahbaikan SKS dan MMAS yang ada dengan memperkenalkan satu parameter pereputan dinamik $(1 - p[r(r, s)])$ atau lebih dikenali sebagai pembolehubah ruapan (Li & Gong 2003). Pembolehubah ini digunakan untuk mengelakkan paras phoremone terlalu tinggi dan mencapai optima setempat (local optima). Secara teorinya kumuhan boleh teruap dengan cepat bila ia aktif (intensifid) dan kurang meruap bila ia lemah (faint). Pembolehubah ruapan ini diaplikasikan di kedua-dua pengemaskinian kumuhan menggunakan peraturan kemasikini phoremone tempatan (persamaan (8)) dan peraturan kemasikini phoremone global (persamaan (9)) yang berikut.

$$\tau(r,s) \leftarrow (1 - p[\tau(r,s)]) \tau(r,s) + p \cdot \Delta\tau_0 \quad (8)$$

dengan $\Delta\tau_0 = (1/L_{mn} \cdot n)$

Li dan Gong juga, mencuba untuk mempercepatkan penyelesaian secara pengiraan dengan membenarkan laluan terbaik dan terburuk yang dilalui oleh semut untuk membenarkan kemaskini kumuhan. L_{gw} adalah jarak laluan global yang terlemah dan L_{gb} adalah jarak laluan terbaik dari tempat mula lawatan.

$$\tau(r,s) \leftarrow (1 - p[\tau(r,s)]) \tau(r,s) + C \quad (9)$$

di mana

$$C = \begin{cases} p \cdot \Delta\tau(r,s) & \text{if } (r,s) \in \text{Perjalanan Global Terbaik} \\ -p \Delta\tau(r,s) & \text{if } (r,s) \in \text{Perjalanan Global Terburuk} \\ 0 & \text{if } (r,s) \in \text{Selainnya} \end{cases}$$

dan

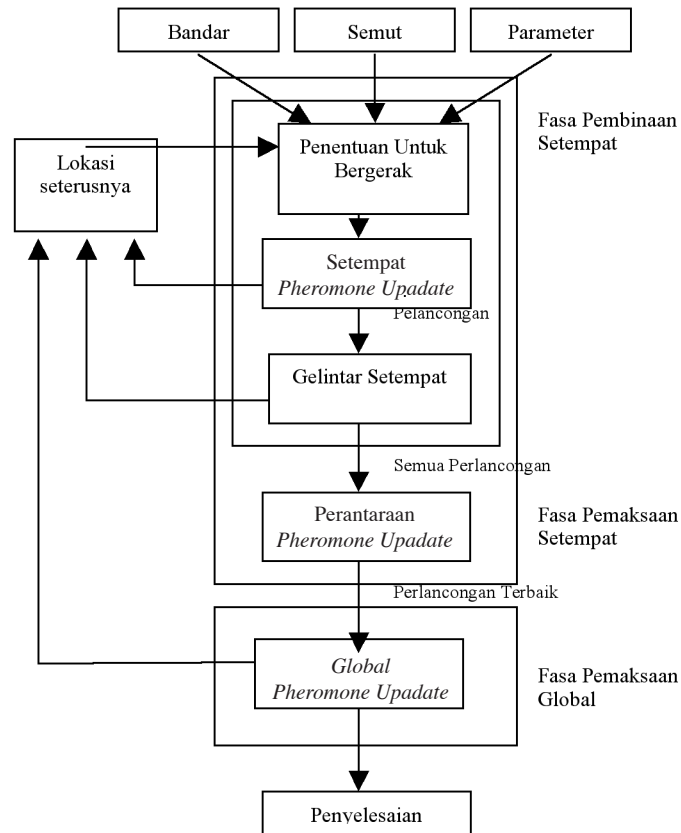
$$\Delta(r,s) = \begin{cases} (L_{gb})^{-1} \\ (L_{gw})^{-1} \end{cases}$$

Kualiti penyelesaian dan pelaksanaan dalam pengoptimuman algoritma koloni semut dapat dipertingkatkan menggunakan pelbagai strategik (Gendreau & Potvin 2005) seperti senarai calon, jangan lihat pun atau menggunakan pelbagai kaedah heuristik.

SISTEM KOLONI SEMUT DINAMIK TIGA PARAS KEMASKINI (SKSD3)

SKSD3 mengambil kira konsep asas koloni semut dalam SKS dan SKSD. Walaubagaimana pun, kajian ini mempertimbangkan perlakuan individual semut yang ditunjukkan dalam Rajah 2. Peranan individual semut diambil kira dalam mengenalpasti penyelesaian terbaik dengan cara mengemaskini paras phoremone. Jadi, SKSD3 berbeza dengan algoritma yang lain dalam 3 keadaan. Pertamanya, mengambil semua pengetahuan dalam semua kumpulan semut dan mengemakini paras phoremone bilamana pengetahuan daripada semua kumpulan diketahui. Cara ini akan mengurangkan kadar phoremone bertambah secara cepat dan juga berkemungkinan dapat mempercepatkan proses untuk mendapatkan penyelesaian terbaik. Kedua, menggunakan penalti yang dinamik untuk pilihan laluan yang buruk. Ini akan membuka peluang kepada semut untuk mengarah, menghadkan kehendak dan menyediakan amaran kepada keputusan semut untuk pergerakan seterusnya. Ketiganya, hanya satu sahaja pilihan laluan terbaik dalam kumpulan yang dipertimbangkan, bila mengaplikasikan peraturan kemaskini phoremone global dibandingkan dengan hasil terbaik dari yang terbaik dan laluan terburuk dari yang terbaik. Rajah 3 menunjukkan aliran kerja model SKSD3.

Pada fasa pembaikan setempat, semua bandar diambil kira bermula di bandar r. Setiap semut akan membuat laluan yang lengkap dan keputusan untuk memilih pergerakan ke bandar s adalah menggunakan peraturan keadaan peralihan persamaan (1) dan persamaan (3). Setiap kali seekor semut melawati bandar s, ia akan mengubahsuai paras phoremone dengan menggunakan peraturan kemaskini phoremone setempat persamaan (4). Para penyelidik dalam bidang ini beranggapan bahawa bila seekor semut mendatangi lokasi baru maka ia akan membuatkan deposit dan ruapan pada phoremone yang sama.



RAJAH 3. Aliran Kerja SKSD3

Apabila semua semut dalam kumpulan telah menamatkan laluanannya, pengetahuan bagi semua ahli dalam kumpulan kemudiannya digunakan untuk mengubah paras phoremone menggunakan peraturan kemaskini phoremone perantaraan yang ditunjukkan dalam persamaan (10). Peraturan ini dilakukan di fasa pemaksaan. Kemaskini ini diperlukan sebelum semua semut dalam kumpulan diberikan tugas yang baru. Di fasa ini, parameter pereputan dinamik digunakan kerana dapat membantu mengelakkan keadaan tepu yang awal, dan dapat mengurangkan kebarangkalian paras phoremone bertambah ketahap terlalu tinggi.

$$\tau(r,s) \leftarrow (1 - [p \cdot \tau(r,s)]) \tau(r,s) + \Delta C$$

di mana

$$\Delta C = \begin{cases} p \cdot \Delta \tau(r,s) & \text{if } (r,s) \in \text{Laluan Global Terbaik} \\ -[p \cdot \tau(r,s)] \Delta \tau(r,s) & \text{if } (r,s) \in \text{Laluan Global Terburuk} \\ 0 & \text{if } (r,s) \in \text{Selainnya} \end{cases} \quad (10)$$

dan

$$\Delta \tau(r,s) = \begin{cases} (L_{grb})^{-1} \\ (L_{grw})^{-1} \end{cases}$$

Semua perlancongan semasa dalam kumpulan dibandingkan dengan kumpulan perlancongan terbaik di dalam lelaran semasa dan kumpulan perlancongan terburuk daripada mula cubaan perlancongan. Jika tidak ada nilai yang padan diperolehi, penjuru tersebut disetkan sebagai peruapan dinamik yang normal. Kaedah ini akan meningkatkan keupayaan semut untuk menghasilkan perlancongan terbaik tetapi melembapkan perlancongan terburuk daripada kumpulan yang baik. Penalti dinamik $[p.\tau(r,s)]$ digunakan untuk membuat amaran kepada semua semut agar jangan memilih laluan yang buruk. L_{gb} ialah jumlah jarak perlancongan terbaik dalam lelaran semasa, dan L_{gw} ialah jumlah jarak perlancongan terburuk dalam kumpulan daripada awal perlancongan.

Paras phoremone seterusnya dikemaskini menggunakan peraturan kemaskini phoremone global, persamaan (11). Hanya perlancongan terbaik dalam kumpulan yang dipertimbang untuk kemaskini paras phoremone. Setiap pergerakan perlancongan akan dibanding dengan setiap pergerakan dalam perlancongan yang lengkap iaitu perlancongan terbaik dari yang terbaik dan terburuk dari yang terbaik. Kaedah ini membantu gelintaran yang lebih baik dalam usaha untuk mendapatkan penyelesaian terbaik.

$$\tau(r,s) \leftarrow (1 - [p.\tau(r,s)]) \tau(r,s) + \Delta C$$

di mana

$$\Delta C = \begin{cases} p.\Delta\tau(r,s) & \text{if } (r,s) \in \text{Laluan Global Terbaik} \\ -[p.\tau(r,s)]\Delta\tau(r,s) & \text{if } (r,s) \in \text{Laluan Global Terburuk} \\ 0 & \text{if } (r,s) \in \text{Selainnya} \end{cases} \quad (11)$$

dan

$$\Delta\tau(r,s) = \begin{cases} (L_{gb})^{-1} \\ (L_{gw})^{-1} \end{cases}$$

L_{gb} ialah jumlah jarak laluan terbaik global (terbaik daripada terbaik), L_{gw} ialah jumlah jarak perlancongan terburuk global daripada mula bergerak. Rajah 4, menunjukkan bagaimana algoritma SKSD3 ini berfungsi. Algoritma ini adalah mewakili algoritma asal SKSD3, tanpa mengaplikasikan sebarang teknik pengoptimuman dan heuristik.

PENYEDIAAN UJIKAJI

Algoritma ini diuji menggunakan pelbagai set data yang terdapat dalam TSPLIB (TSPLIB95,2009). Algoritma ini dibangunkan dalam bahasa pengaturcaraan C, dalam persekitaran Microsoft Visual Studio. Ujian dijalankan menggunakan komputer pemrosesan 1.86GHz dengan 1G ingatan. Ujian menggunakan kaedah bundar atas normal ($\geq 0.5 = 1$) untuk mengira jarak dalam integer. Pelbagai saiz populasi set data telah diuji untuk memastikan jumlah semut terbaik dalam data tersebut. Terdapat 2 jenis set data yang diuji: Jarak Euclidean (Oliver30, Berlin52, Kroc100) dan jarak GEO (Burma14).

Ujikaji mengenalpasti algoritma yang mencapai jarak optima yang terbaik boleh dicapai. Jika semua data algoritma mencapai jarak terbaik, maka ujian kelajuan menjadi ukuran kedua. Ujikaji mencerap 5 jenis pengukuran untuk tujuan perbandingan. Lajur pertama merekod jarak tanda aras yang patut diperolehi sebagai penyelesaian. Baris kedua merekod jarak terbaik daripada mula perjalanan berbanding dengan jarak tanda aras. Baris ketiga menunjukkan jumlah lelaran yang diperlukan untuk menghasilkan jarak terbaik. Baris keempat merekod masa terbaik diperolehi daripada 15 kali cubaan. Baris kelima pula merekod purata masa untuk mendapat penyelesaian dengan 15 kali cubaan. Jarak diukur dalam nilai Euclidean dan GEO. Jarak

```

GlobalBestTour = ∞;
GlobalWorstTour = 0;
LocalGroupWorstTour = 0;
Initialize pheromone level for all cities =  $\tau_0$ ;
Generate initial solution using Nearest Neighbor (NN) heuristic;
CPU timer starts;
/* Trial begins */
Do
  /* Iteration begins */
  If i <= n
    LocalGroupBestTour = ∞;
    For k = 1 to m
      Start city = i;
      Do
        Select the next city j;
        /*Perform Local Pheromone Update*/
        Update trail level  $\tau_{ij}$ ; (Equation (4))
      While (until all cities visited)
    EndFor
    /*Perform Intermediate Pheromone Update*/
    For k = 1 to m
      Compute tour distance;
      If (tour distance < LocalGroupBestTour)
        LocalGroupBest = current solution;
      Else if (tour distance > LocalGroupWorstTour)
        LocalGroupWorst = current solution;
      Else
        /*Pheromone updates for others*/
        Update trail level  $\tau_{ij}$ ; (Equation (10))
      EndIf
    /*Pheromone updates for LocalGroupBest & LocalGroupWorst*/
    Update trail level  $\tau_{ij}$  for LocalGroupBest; (Equation (10))
    Update trail level  $\tau_{ij}$  for LocalGroupWorst; (Equation (10))
    EndFor
  EndIf
  /*Perform Global Pheromone Update*/
  Compute tour distance of LocalGroupBest
  If (tour distance < GlobalBestTour)
    GlobalBest = LocalGroupBest;
  Else if (tour distance > GlobalWorstTour)
    GlobalWorst = LocalGroupBest;
  Else
    /*Pheromone updates for others*/
    Update trail level  $\tau_{ij}$ ; (Equation (11))
  EndIf
  /*Pheromone updates for GlobalBest & GlobalWorst*/
  Update trail level  $\tau_{ij}$  for GlobalBest; (Equation (11))
  Update trail level  $\tau_{ij}$  for GlobalWorst; (Equation (11))
While (until all termination statements satisfied)

```

RAJAH 4. SKSD3 Algoritma

sebenarnya dapat dikira dengan membuat pengiraan ke atas nilai jarak dalam bentuk Euclidean. Manakala jarak nilai integer diguna sebagai asas pengiraan jarak untuk set data GEO.

Jadual 1 menunjukkan set parameter terbaik untuk ujikaji SKSD3. Set ini sama dengan penyediaan ujikaji yang dijalankan oleh algoritma SKS dan SKSD sebelum ini (Fleischer 2003; Merloti 2004; Grosan & Abraham 2006; Rais et. al. 2007; Dorigo & Socha 2007).

JADUAL 1. Penyediaan Parameter Ujikaji

Parameters	Value
Ants Population Size	10
q_0	0.9
β	2
p	0.1
Max Iterations	10000

KEPUTUSAN DAN PERBINCANGAN

Keberkesanan SKSD3 dilihat dari beberapa aspek: jarak terdekat diperoleh, prestasi gelintar (menggunakan graf log) dan ujian signifikan. Dua kaedah ujian signifikan dijalankan iaitu menggunakan kaedah statistik iaitu ujian t dan ujian ROC.

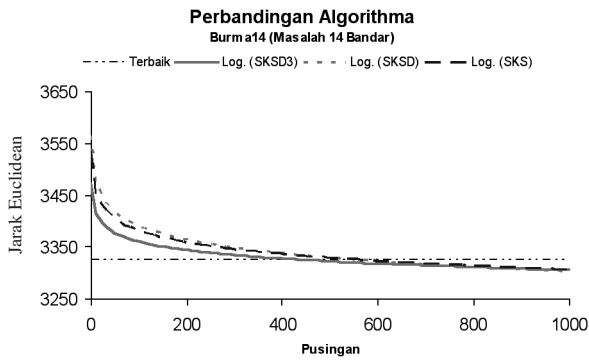
Jadual 2 menunjukkan keputusan ujikaji SKSD3 berbanding algoritma yang lain. Daripada analisis jadual tersebut algoritma SKSD3 berupaya mencapai jarak tanda aras yang optima untuk semua set data kecuali KroC100, walaupun bagaimanapun perbezaan terlalu kecil (0.6%). SKS dan SKSD hanya berupaya mencapai jarak terdekat tanda aras, tetapi hanya untuk set data Oliver30 dan Burma14 sahaja. SKS berupaya mencapai 2.5% lebih panjang berbanding dengan jarak tanda aras untuk data Berlin52 dan berupaya mencapai 4.4% lebih panjang berbanding dengan jarak tanda aras dan untuk KroC100. Manakala SKSD berupaya mencapai 2.0% lebih panjang berbanding dengan jarak tanda aras untuk Berlin52 dan berupaya mencapai 2.1% lebih panjang berbanding dengan jarak tanda aras untuk data KroC100. Jelas disini menunjukkan algoritma SKSD3 lebih baik berbanding SKSD dan SKS kerana keupayaannya menghasilkan penyelesaian jarak terdekat yang lebih bagi semua set data. Dari sudut masa pelaksanaan pula, SKSD3 berjaya mengurangkan masa untuk mendapatkan penyelesaian terbaik sebanyak 90% berbanding SKS untuk data Oliver30 dan 75% berbanding SKSD untuk data Burma14. Walau bagaimanapun, keupayaan SKSD3 agak berkurangan untuk set data yang besar.

Prestasi gelintar boleh dilihat menggunakan log graf jarak terdekat yang dicapai melawan lelaran. Rajah 7(a-d) menunjukkan perbandingan prestasi gelintaran antara SKSD3 dengan SKS dan SKSD untuk set data Burma14, Oliver30, Berlin30 dan KroC100. Rajah menunjukkan bahawa SKSD3 mempunyai prestasi gelintaran yang baik untuk set data yang kecil (Burma14 dan Oliver30) berbanding algoritma yang lain. Manakala untuk set data besar, prestasi gelintar SKSD3 agak lambat diawal gelintar tetapi menunjukkan prestasi yang baik dipertengahan dan diakhir pelaksanaan.

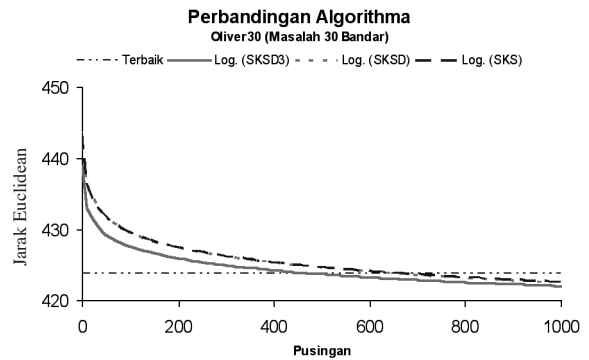
Jadual 3 menunjukkan hasil ujian signifikan algoritma SKSD3 terhadap algoritma lain menggunakan analisis statistik ujian t. Manakala Jadual 4 pula memapar hasil ujian signifikan algoritma SKSD3 berbanding algoritma yang lain menggunakan menggunakan ujian ROC.

JADUAL 2. Hasil Ujikkaji Jarak Terdekat dan Masa Pelaksanaan Algoritma SKSD3, SKSD dan SKS

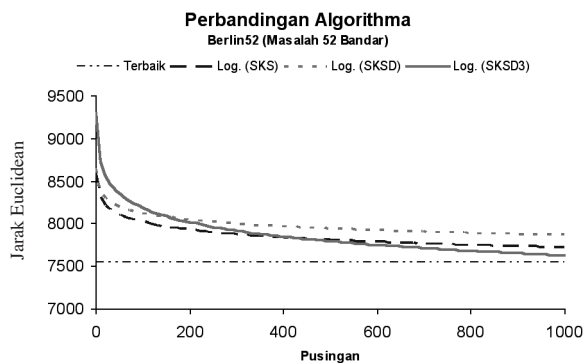
TSP Problem	Benchmark Distance	SKSD3 ($\beta=2, q_0=0.9, p=0.1$) – m=10				SKSD ($\beta=2, q_0=0.9, p=0.1$) – m=10				SKS ($\beta=2, q_0=0.9, p=0.1$) – m=10			
		Best Distance	Best Iteration	Best Time (Sec)	Average Time (Sec)	Best Distance	Best Iteration	Best Time (Sec)	Average Time (Sec)	Best Distance	Best Iteration	Best Time (Sec)	Average Time (Sec)
Burma14 (14-city problem)	3323 (N/A)	94	0.953	1.003	3323 (3330.61)	527	4.766	4.823	3323 (3330.61)	527	4.703	4.767	
Oliver30 (30-city problem)	420 (423.74)	168	7.000	7.043	420 (423.74)	1826	71.484	71.582	420 (423.74)	1826	72.391	72.501	
Berlin52 (52-city problem)	7542 (N/A)	6447	1278.407	1278.590	7881 (7880.78)	3661	694.594	694.835	7732 (7734.11)	6050	1164.766	1165.081	
KroC100 (100-city problem)	20749 (N/A)	7954	10854.407	10856.385	21190 (21191.37)	1720	2240.984	2242.229	21170 (21172.26)	2670	3531.782	3532.969	



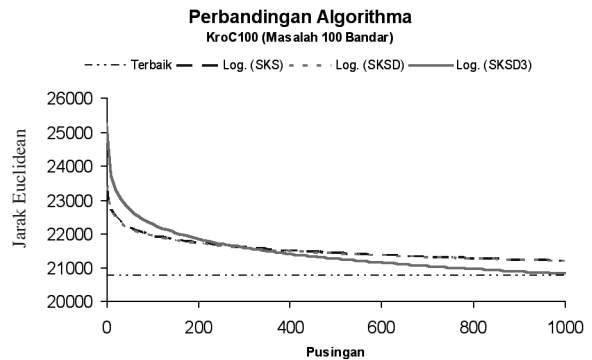
RAJAH 7a. Graf log perbandingan antara algoritma untuk Burma14



RAJAH 7b. Graf Log perbandingan antara algoritma untuk Oliver30



RAJAH 7c. Graf log perbandingan antara algoritma untuk Berlin52



RAJAH 7d. Graf log perbandingan antara algoritma untuk KroC100

Daripada jadual 3 dan 4 mendapati bahawa algoritma SKSD3 telah menunjukkan hasil perbezaan yang signifikan di antara kedua algoritma SKSD dan SKS samada menggunakan kaedah ujian t dan ujian ROC

KESIMPULAN DAN CADANGAN

Hasil daripada keputusan ujikaji dinyatakan sebelum ini, dapat disimpulkan bahawa memanipulasikan dan memperhebatkan pengetahuan kepada individual semut yang dibenamkan dalam sistem koloni semut, mempunyai kebaikan dalam menghasilkan penyelesaian masalah terbaik domain PPJ. Memanfaatkan pengetahuan pada setiap semut dapat mempercepat proses untuk mendapatkan makanan atau menghasilkan penyelesaian terbaik. SKSD3 telah berjaya memperoleh penyelesaian jarak terdekat yang terbaik berbanding algoritma yang lain. SKSD3 juga mempunyai kebolehan mengelintar dengan baik untuk set data kecil tetapi kurang baik untuk set data yang besar. Walau bagaimanapun, keupayaan SKSD3 boleh dipertingkatkan lagi dengan mengaplikasikan pelbagai jenis strategik teknik yang dilakukan oleh penyelidikan sebelum ini iaitu senarai calon, pelicinan laluan phoremone dan mengaplikasikan konsep semut ‘elitise’.

JADUAL 3. Ujian Pasangan T-test di antara algoritma SKSD3, SKSD and SKS

Set Data	Pasangan Algoritma Ujian	Purata	Sisihan Piawai	Purata Ralat Piawai	95% Perbezaan Keyakinan		t	df	Sig (2-ekor)
					Aras Bawah Aras Atas				
					Aras Bawah	Aras Atas			
Burma14	SKSDE-SLS	-0954	6.565	0.066	-1.082	-0.825	-14.528	10000	0.000
	SLSD3-SLSD	-1.037	6.164	0.062	-1.158	-0.916	-16.828	10000	0.000
	SLSDE-SKS	-0.196	0.626	0.006	-0.208	-0.184	-31/383	10000	0.000
Oliver30	SKSD3-SLSD	-0189	0.552	0.005	-0.200	-0.179	-34.366	10000	0.000
	SKSD3-SKS	-88.263	98.034	0.980	-90.185	-86.341	-90.038	10000	0.000
Berlin52	SKSD3-SKSD	-222.806	105.864	1.058	-224.881	-220.731	-210.474	10000	0.000
	SKSD3-SKS	684.625	1477.471	14.773	655.665	713.585	46.340	10000	0.000
Pr76	SKSD3-SKSD	-1039.128	1410.903	14.108	-1066.783	-1011.473	-73.654	10000	0.000
	SKSD3-SKS	-267.177	150.677	1.506	-270.131	-264.224	-177.327	10000	0.000
KroC100	SKSD3-SLSD	-250.524	138.710	1.387	-253.243	-247.805	-180.619	10000	0.000

JADUAL 4. Ujian Signifikan menggunakan ROC diantara SKSD3, SKSD and SKS

Set Data	Pasangan Algorithma	Keluasan	Ralat Piawai	95% berasimptot		Signifikan berasimptot
				Julat Bawah	Keyakinan Atas	
Burma14	SKSD3-SKS	0.589	0.014	0.561	0.617	0.000
	SKSD3-SKSD	0.589	0.014	0.561	0.617	0.000
Oliver30	SKSD3-SKS	0.546	0.008	0.531	0.561	0.000
	SKSD3-SKSD	0.546	0.008	0.531	0.561	0.000
Berlin52	SKSD3-SKS	0.981	0.002	0.978	0.985	0.000
	SKSD3-SKSD	0.823	0.004	0.816	0.831	0.000
Pr76	SKSD3-SKS	0.983	0.002	0.980	0.986	0.000
	SKSD3-SKSD	0.996	0.000	0.995	0.997	0.000
KroC100	SKSD3-SKS	0.897	0.004	0.889	0.904	0.000
	SKSD3-SKSD	1.000	0.000	0.999	1.000	0.000

RUJUKAN

- Dorigo, M., Di Caro, G and Gambardella, L. M. 1999. Ant Algorithms for Discrete Optimization. *Journal of Artificial Life* 5(2): 137-172.
- Fleischer, M. 2003. Foundation of Swarm Intelligence: From Principles to Practice, Conference on Swarming: Network Enabled C4ISR.
- Dorigo, M., Maniezzo, V. and Coloni, A. 1996 The Ant System: Optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Man and Cybernetics-Part B* 26(1): 1-13.
- Eric Bonabeau & Christopher Meyer. 2001. Swarm Intelligence: A Whole New Way to Think About Business. *Harvard Business Review*. pp: 107-114. Harvard Business School Publishing Corporation.
- Grosan, C. and Abraham, A. 2006. Stigmergic Optimization: Inspiration, Technologies and Perspectives. *Studies in Computational Intelligence Journal*, Vol 31, Springer Publishing.

- Dorigo, M and Socha, K. 2006. An Introduction to Ant Colony Optimization, Technical Report IRIDIA 2006-010.
- Dorigo, M. and Caro, G. D. 1999. The Ant Colony Optimization Meta-Heuristic, In D. Corne, M. Dorigo and F. Glover, editors, *New Ideas in Optimization*, McGraw-Hill: 11-32.
- Deneubourg, J.L., Aron, S., Goss, S. and Pasteels, J. M. 1990. The self-organizing exploratory pattern of Argentine Ant, *Journal of Insect Behavior* 3:159 – 168.
- Dorigo, M., Bonabeau, E. and Theraulaz, G. 2000. Ant algorithms and stigmergy, *Journal of Future Generation Computer Systems* 16: 851- 871.
- Anderson, C. and Franks, N. R. 2000. Teams in animal societies, *Journal of Behavioral Ecology* 12(5): 534-540.
- Rais, H. M., Othman, Z. A. and Hamdan, A. R. 2007. Improved Dynamic Ant Colony System (SKSD) on Symmetric Traveling Salesman Problem (TSP), International Conference on Intelligent and Advanced System.
- Dorigo, M. and Gambardella, L. M. 1997. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem, *IEEE Transaction of Evolutionary Computation* 1(1): 53-66.
- Gaertner, D. and Clark, K. 2005. On Optimal Parameters for Ant Colony Optimization algorithms, Proceedings of International Conference on Artificial Intelligent (ICAI2005)1.
- Dorigo, M. and Socha, K. 2007. An Introduction to Ant Colony Optimization, Book Chapter in *Approximation Algorithms and Metaheuristic*, CRC Press 2007.
- Dorigo, M., Birattari, M. and Stützle, T. 2006. Ant Colony Optimization – Artificial Ants as a Computational Intelligence Technique, *IEEE Computation Intelligent Magazine*.
- Stützle, T. and Hoos, H. H. 2000. MAX-MIN Ant System, *Journal of Future Generation Computer Systems* 16(8): 889-914.
- Merloti, P. E. 2004. Optimization Algorithms Inspired by Biological Ants and Swarm Behavior, San Diego State University, Artificial Intelligence Technical Report, CS550, San Diego.
- Li, Y. and Gong, S. 2003. Dynamic ant colony optimization for TSP, *Int J Adv Manuf Technol* 22: 528-533.
- Gendreau, J and Potvin, Y. 2005. Metaheuristic in Combinatorial Optimization, *Annals of Operation Research* 140: 189-213.
- TSPLIB95, 1995. <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp/> (20hb Jun 2009)

Zulaiha Ali Othman, Mohd Helmi Rais, Roslan Ahmad & Abdul Razak Hamdan
 Jabatan Sains dan Pengurusan Sistem
 Fakulti Teknologi dan Sains Maklumat
 Universiti Kebangsaan Malaysia
 zao@ftsm.ukm.my
 Mubin7677@yahoo.com
 Roslan.dr@gmail.com
 arh@ftsm.ukm.my