

**MENYELESAIKAN MASALAH  
PERANCANGAN JUJUKAN PEMASANGAN  
MENGUNAKAN ALGORITMA PENAPIS  
KALMAN DISELAKUKAN**

**AINIZAR BINTI MUSTAPA**

**UMP**

**SARJANA SAINS**

**UNIVERSITI MALAYSIA PAHANG**

## UNIVERSITI MALAYSIA PAHANG

### PENGISYTIHARAN TESIS DAN HAKCIPTA

Nama Penuh Pengarang : AINIZAR BINTI MUSTAPA

Tarikh Lahir : 10 OKTOBER 1983

Tajuk : MENYELESAIKAN MASALAH PERANCANGAN JUJUKAN  
PEMASANGAN MENGGUNAKAN ALGORITMA PENAPIS  
KALMAN DISELAKUKAN

Sesi Akademik : SEMESTER 2 2019/2020

Saya mengakui bahawa tesis ini dikelaskan sebagai:

- SULIT (Mengandungi maklumat rahsia di bawah Akta Rahsia Rasmi 1997)\*
- TERHAD (Mengandungi maklumat yang dibatasi sebagaimana yang ditentukan oleh organisasi di mana kajian telah dilakukan)\*
- AKSES TERBUKA Saya bersetuju bahawa tesis saya akan diterbitkan sebagai akses terbuka dalam talian (Teks Penuh)

Saya mengakui bahawa Universiti Malaysia Pahang berhak seperti berikut:

1. Tesis ini adalah Hak Milik Universiti Malaysia Pahang
2. Perpustakaan Universiti Malaysia Pahang berhak membuat salinan tesis untuk tujuan kajian sahaja.
3. Perpustakaan mempunyai hak untuk membuat salinan tesis untuk pertukaran akademik.

Disahkan oleh:

\_\_\_\_\_  
(Tandatangan Pelajar)

831010-06-5500

\_\_\_\_\_  
MyKad/Nombor Passport  
Tarikh:

\_\_\_\_\_  
(Tandatangan Penyelia)

Prof. Madya Dr. Zuwairie bin Ibrahim

\_\_\_\_\_  
Nama Penyelia  
Tarikh:

NOTA: \* Jika tesis SULIT atau TERHAD, sila lampirkan surat perisytiharan tesis.

## **PENGAKUAN PENYELIA**

Kami dengan ini mengisytiharkan bahawa kami telah menyemak tesis ini dan pada pendapat kami, tesis ini adalah memadai dari segi skop dan kualiti untuk penganugerahan Sarjana Sains.

---

(Tandatangan Penyelia)

Nama Penuh : PROF. MADYA. DR. ZUWAIKIE BIN IBRAHIM

Jawatan : PROFESOR MADYA

Tarikh :

---

(Tandatangan Penyelia Bersama)

Nama Penuh : ZULKIFLI BIN MD. YUSOF

Jawatan : PENSYARAH KANAN

Tarikh :

## **PENGAKUAN PELAJAR**

Saya dengan ini mengisytiharkan bahawa kerja dalam tesis ini adalah berdasarkan kerja asal saya kecuali petikan yang telah diakui dengan sewajarnya. Saya juga mengisytiharkan bahawa ia tidak sebelum ini atau serentak diserahkan untuk ijazah lain di Universiti Malaysia Pahang atau mana-mana institusi lain.

---

(Tandatangan Pelajar)

Nama Penuh : AINIZAR BINTI MUSTAPA

Nombor Pelajar : MMF15009

Tarikh :



UMP

MENYELESAIKAN MASALAH PERANCANGAN JUJUKAN PEMASANGAN  
MENGUNAKAN ALGORITMA PENAPIS KALMAN DISELAKUKAN



AINIZAR BINTI MUSTAPA

Tesis yang dikemukakan sebagai memenuhi keperluan  
untuk penganugerahan Ijazah Sarjana Sains

UMP

Kolej Kejuruteraan

UNIVERSITI MALAYSIA PAHANG

JULAI 2020

## PENGHARGAAN

Syukur Alhamdulillah ke hadrat Allah S.W.T. kerana di atas limpah dan kurniaNya, tesis ini berjaya disiapkan walaupun menempuhi pelbagai dugaan dan rintangan. Selawat dan salam ke atas junjungan besar Nabi Muhammad S.A.W. yang diutuskan Allah sebagai *rahmatan lil 'alamin*, guru dan contoh tauladan terbaik untuk manusia sepanjang zaman.

Saya ingin mengucapkan jutaan terima kasih kepada Prof. Madya Dr. Zuwairie bin Ibrahim dan Encik Zulkifli bin Md. Yusof, selaku penyelia dan penyelia bersama atas kesabaran, sokongan, nasihat dan bimbingan untuk kejayaan penghasilan tesis ini. Tidak dilupakan ribuan terima kasih kepada Dr. Ismail bin Ibrahim serta rakan-rakan seperjuangan kerana memberi inspirasi dan dorongan di sepanjang pengajian.

Terima kasih kepada pihak Kementerian Pengajian Tinggi Malaysia kerana telah menganugerahkan saya biasiswa MyBrain15 bagi melanjutkan pelajaran ke peringkat Sarjana. Terima kasih juga kepada Kolej Kejuruteraan UMP, Kolej Teknologi Kejuruteraan (Fakulti Teknologi Kejuruteraan Pembuatan dan Mekatronik) UMP, serta Institut Pengajian Pasca Siswazah UMP atas panduan dan sokongan penuh terhadap kajian ini. Terima kasih kepada Isuzu Hicom Malaysia Sdn. Bhd. kerana memahami dan menyokong cita-cita saya sebagai seorang kakitangan syarikat yang berhasrat menyambung pelajaran ke peringkat Sarjana.

Jutaan penghargaan dan terima kasih kepada ibu (Puan Paridah), ayah (Allahyarham Mustapa), suami (Encik Rismayuddin), anak-anak (Aisyah Rayhana, Aminah Raysha dan Allahyarham Ahmad Raykarl) dan seluruh keluarga atas doa, kesabaran, sokongan, toleransi, cinta dan kasih sayang kalian, sehingga tesis dan pengajian ini berjaya disempurnakan.

Akhir kata, ucapan terima kasih ini ditujukan kepada semua yang terlibat secara langsung dan tidak langsung dalam memberikan sumbangan cadangan dan bantuan dalam menyiapkan tesis ini. Semoga kajian dan tesis ini dapat dijadikan wadah ilmu yang berguna untuk tatapan generasi akan datang.

## ABSTRAK

Perancangan jujukan pemasangan (*Assembly Sequence Planning* - ASP) memainkan peranan penting dalam reka bentuk dan pembuatan produk. Jujukan pemasangan mempengaruhi keseluruhan produktiviti kerana ia menentukan kepantasan dan ketepatan produk itu dipasang. Objektif utama ASP adalah untuk menentukan jujukan pemasangan komponen untuk memendekkan masa pemasangan atau menjimatkan kos pemasangan. Walau bagaimanapun, ASP juga dikenali sebagai masalah pengoptimuman gabungan klasik yang sukar. Dengan peningkatan bilangan komponen bagi sesuatu produk, ASP menjadi lebih sukar dan algoritma berasaskan grafik tradisional tidak dapat menyelesaikannya dengan berkesan. Terdapat pelbagai metaheuristik yang wujud pada masa kini. Walau bagaimanapun, tidak semua metaheuristik dibangunkan untuk beroperasi di ruang carian diskret. Salah satu contoh algoritma metaheuristik ialah Kalman. Maka, bagi tujuan menyelesaikan masalah pengoptimuman gabungan (*Combinatorial Optimization Problem* - COP) yang diskret menggunakan metaheuristik serta menilai prestasi algoritma yang dicadangkan, satu kajian kes ASP telah dijalankan. Prestasi algoritma penapis Kalman diselakukan (*Simulated Kalman Filter* - SKF) lanjutan yang dinamakan penapis Kalman diselakukan binari (*Binary Simulated Kalman Filter* – BSKF), penapis Kalman diselakukan dimodulasi sudut (*Angle Modulated Simulated Kalman Filter* – AMSKF), dan penapis Kalman diselakukan dinilai jarak (*Distance-Evaluated Simulated Kalman Filter* - DESKF) dibandingkan dengan hasil kajian lalu yang menggunakan algoritma carian graviti binari (*Binary Gravitational Search Algorithm* - BGSA), algoritma pengoptimuman kerumunan zarah binari (*Binary Particle Swarm Optimization* - BPSO), algoritma carian graviti berbilang keadaan (*Multi-State Gravitational Search Algorithm* - MSGSA), algoritma carian graviti berbilang keadaan dengan peraturan tertanam (*Multi-State Gravitational Search Algorithm with an Embedded Rule* - MSGSAER), algoritma pengoptimuman kerumunan zarah berbilang keadaan (*Multi-State Particle Swarm Optimization* - MSPSO), dan algoritma pengoptimuman sekawan zarah berbilang keadaan dengan peraturan tertanam (*Multi-State Particle Swarm Optimization with an Embedded Rule* - MSPSOER) dalam menyelesaikan masalah ASP. Dengan menggunakan satu kajian kes ASP, hasil eksperimen menunjukkan AMSKF mengatasi BSKF, DESKF dan enam algoritma lain daripada kajian lalu dengan kelebihan sehingga 0.95% dalam mencari penyelesaian yang optimum.

## ABSTRACT

Assembly sequence planning (ASP) plays an important role in the product design and manufacturing. Assembly sequence influences overall productivity because it determines how fast and accurate the product is assembled. One of the main objective of ASP is to determine the sequence of component installation to shorten assembly time or save the assembly costs. However, ASP is also known as a classical hard combinatorial optimization problem. With the increasing of the quantity of product components, the ASP becomes more difficult and the traditional graph-based algorithm cannot solve it effectively. There are various metaheuristics exist in literature nowadays. However, not all metaheuristics were originally developed to operate in discrete search space. Example of metaheuristics algorithm is Kalman. In order to solve discrete combinatorial optimization problems (COPs) using metaheuristics, and evaluate the performances of the proposed algorithms, a case study of ASP is conducted. The performance of the extended Simulated Kalman Filter (SKF) named Binary Simulated Kalman Filter (BSKF), Angle Modulated Simulated Kalman Filter (AMSKF), and Distance Evaluated Simulated Kalman Filter (DESKF) are compared against previous studies which applied the Binary Gravitational Search Algorithm (BGSA), the Binary Particle Swarm Optimization (BPSO), the Multi-State Gravitational Search Algorithm (MSGSA), the Multi-State Gravitational Search Algorithm with an Embedded Rule (MSGSAER), the Multi-State Particle Swarm Optimization (MSPSO), and the Multi-State Particle Swarm Optimization with an Embedded Rule (MSPSOER) in solving ASP problem. Using a case study of ASP, the experimental results showed the AMSKF outperformed the BSKF, the DESKF and the six other approaches from previous studies by up to 0.95% in finding the optimal solutions.

The logo for UMP (Universiti Malaysia Perlis) is a large, stylized letter 'V' shape. The left side of the 'V' is light blue, the right side is light green, and the bottom point is a darker blue. The letters 'UMP' are written in white, bold, sans-serif font across the center of the 'V'.



# KANDUNGAN

<b>PENGAKUAN</b>	
<b>TAJUK</b>	
<b>PENGHARGAAN</b>	ii
<b>ABSTRAK</b>	iii
<b>ABSTRACT</b>	iv
<b>KANDUNGAN</b>	v
<b>SENARAI JADUAL</b>	vii
<b>SENARAI RAJAH</b>	ix
<b>SENARAI SIMBOL</b>	x
<b>SENARAI SINGKATAN</b>	xi
<b>BAB 1 PENGENALAN</b>	1
1.1 Latar Belakang Kajian	1
1.2 Pernyataan Masalah	3
1.3 Objektif Kajian	3
1.4 Skop Kajian	4
1.5 Manfaat Kajian	4
1.6 Ringkasan Tesis	5
<b>BAB 2 KAJIAN LITERATUR</b>	6
2.1 Pengenalan	6
2.2 ASP	6
2.3 Kekangan ASP	7
2.4 Objektif ASP	9
2.5 ASP untuk Mengurangkan Masa Pemasangan	15
2.6 Model Matematik	16

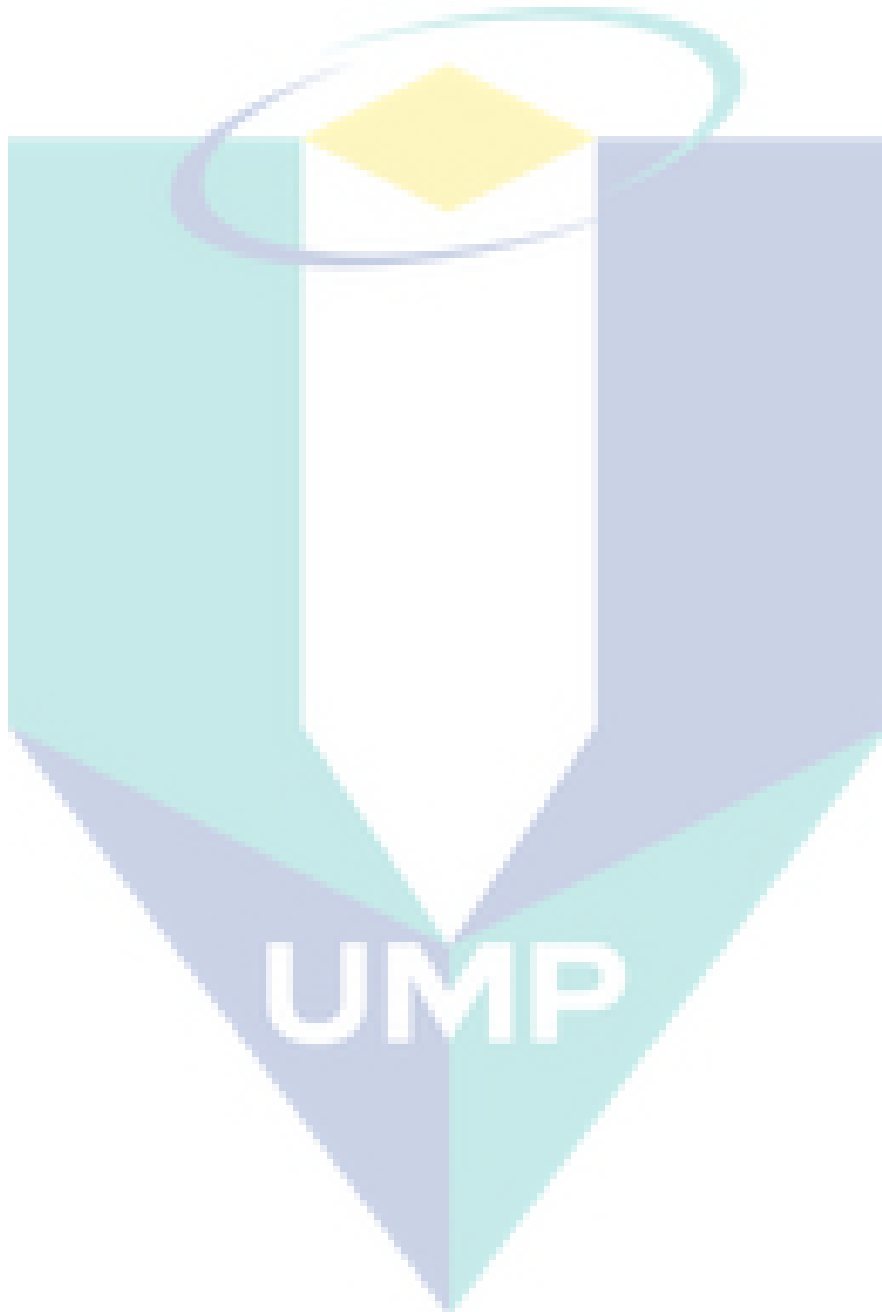
2.7	Pengoptimuman Kawanan Zarah (PSO)	19
2.8	Algoritma Carian Graviti (GSA)	21
2.9	Algoritma Penapis Kalman	24
2.10	Algoritma Penapis Kalman Diselakukan (SKF)	27
2.11	Ringkasan	30
<b>BAB 3 METODOLOGI</b>		<b>31</b>
3.1	Pengenalan	31
3.2	Algoritma Penapis Kalman Simulasi Binari (BSKF)	32
3.3	Algoritma Penapis Kalman Diselakukan Modulasi Sudut (AMSKF)	33
3.4	Algoritma Penapis Kalman Diselakukan Dinilai Jarak (DESKF)	34
3.5	Aplikasi Algoritma SKF Lanjutan dalam ASP	48
3.6	Parameter bagi BSKF, AMSKF, dan DESKF untuk ASP	46
3.7	Ringkasan	46
<b>BAB 4 KEPUTUSAN DAN PERBINCANGAN</b>		<b>47</b>
4.1	Pengenalan	47
4.2	Keputusan Aplikasi BSKF, AMSKF, dan DESKF untuk ASP	48
4.3	Keputusan Aplikasi BSKF, AMSKF, dan DESKF dibandingkan dengan algoritma lain untuk ASP	51
4.4	Ringkasan	56
<b>BAB 5 KESIMPULAN</b>		<b>58</b>
5.1	Kesimpulan	58
5.2	Sumbangan Kajian	59
5.3	Cadangan untuk Masa Hadapan	60
<b>RUJUKAN</b>		<b>62</b>
<b>LAMPIRAN A SENARAI PENERBITAN</b>		<b>77</b>

## SENARAI JADUAL

Jadual 2.1	PM bagi Rajah 2.2	8
Jadual 2.2	Ringkasan kajian ASP menggunakan kaedah pengkomputeran (2000-2016)	12
Jadual 3.1	PM untuk kajian kes	40
Jadual 3.2	CT bagi pelbagai komponen dalam pemasangan	41
Jadual 3.3	Parameter eksperimen untuk pendekatan yang dicadangkan berdasarkan BSKF, AMSKF, dan DESKF dengan 10, 20, 30 agen, untuk 1,000 dan 5,000 lelaran	46
Jadual 4.1	Keputusan untuk kaedah yang dicadangkan berdasarkan kepada BSKF untuk 1,000 lelaran.	48
Jadual 4.2	Keputusan untuk kaedah yang dicadangkan berdasarkan kepada AMSKF untuk 1,000 lelaran.	48
Jadual 4.3	Keputusan untuk kaedah yang dicadangkan berdasarkan kepada DESKF untuk 1,000 lelaran.	48
Jadual 4.4	Keputusan untuk kaedah yang dicadangkan berdasarkan kepada BSKF untuk 5,000 lelaran	49
Jadual 4.5	Keputusan untuk kaedah yang dicadangkan berdasarkan kepada AMSKF untuk 5,000 lelaran	49
Jadual 4.6	Keputusan untuk kaedah yang dicadangkan berdasarkan kepada DESKF untuk 5,000 lelaran	49
Jadual 4.7	Kompilasi keputusan terbaik kaedah yang dicadangkan berdasarkan BSKF, AMSKF, dan DESKF	49
Jadual 4.8	Keputusan terbaik dan jujukan pemasangan yang berkaitan dengan kaedah yang dicadangkan berdasarkan BSKF, AMSKF, dan DESKF	50
Jadual 4.9	Parameter yang digunakan untuk pendekatan yang dicadangkan berdasarkan BPSO, MSPSO, MSPSOER, BGSA, MSGSA dan MSGSAER	52
Jadual 4.10	Keputusan daripada kaedah yang dicadangkan berdasarkan BPSO, MSPSO, MSPSOER, BGSA, MSGSA dan MSGSAER bersama kaedah baru yang dicadangkan berdasarkan BSKF, AMSKF, dan DESKF	53

Jadual 4.11 Keputusan mengikut turutan terbaik dan jujukan pemasangan yang berkaitan dengan pendekatan yang dicadangkan berdasarkan BPSO, MSPSO, MSPSOER, BGSA, MSGSA dan MSGSAER terhadap pendekatan yang dicadangkan berdasarkan BSKF, AMSKF, dan DESKF.

55



## SENARAI RAJAH

Rajah 2.1	Contoh pandangan meletup bagi proses pemasangan	7
Rajah 2.2	PD bagi pemasangan	8
Rajah 2.3	Kekerapan objektif ASP seperti yang diterbitkan dalam kertas kajian	11
Rajah 2.4	Algoritma metaheuristik untuk ASP	14
Rajah 2.5	Algoritma PSO	20
Rajah 2.6	Algoritma GSA	22
Rajah 2.7	Kitaran penapis Kalman yang berterusan	27
Rajah 2.8	Carta alir algoritma SKF	30
Rajah 3.1	Carta alir kaedah penyelidikan yang dijalankan	31
Rajah 3.2	Fungsi pemetaan	32
Rajah 3.3	Carta aliran algoritma BSKF	33
Rajah 3.4	Contoh plot $g(x)$	34
Rajah 3.5	Carta alir algoritma AMSKF	35
Rajah 3.6	Kedudukan ejen. (a) Pada permulaan proses carian (b) Semasa tengah-tengah proses carian (c) Pada akhir proses carian	36
Rajah 3.7	Carta alir algoritma DESKF	38
Rajah 3.8	Contoh jujukan pemasangan yang diwakili oleh zarah	38
Rajah 3.9	PD untuk kajian kes	39
Rajah 3.10	Proses pembentukan jujukan pemasangan setiap zarah atau ejen yang boleh diperbaiki untuk BSKF	43
Rajah 3.11	Proses pembentukan jujukan pemasangan setiap zarah atau ejen yang boleh diperbaiki untuk AMSKF	44
Rajah 3.12	Proses pembentukan jujukan pemasangan setiap zarah atau ejen yang boleh diperbaiki untuk DESKF	45
Rajah 4.1	Keputusan masa pemasangan minimum (Min) untuk kaedah yang dicadangkan berdasarkan BPSO, MSPSO, MSPSOER, BGSA, MSGSA dan MSGSAER, serta kaedah baru yang dicadangkan berdasarkan BSKF, AMSKF, dan DESKF	54
Rajah 5.1	ASP menggunakan algoritma metaheuristik	61

## SENARAI SIMBOL

$A$	Matrik pertukaran keadaan
$A_a$	Masa pemasangan
$B$	Matrik yang berkaitan dengan input kawalan pilihan kepada keadaan
$\beta$	Pemalar $\beta$
$G$	Pemalar graviti
$H$	Matrik yang mentafsir pemetaan dari vektor keadaan kepada vektor pengukuran
$K_t$	Perolehan Kalman
$P_{t-1}$	Anggaran ko-varian pada masa $t-1$
$P_{t t-1}$	Jangkaan (keutamaan) anggaran ko-varian
$P_t$	Terkini (kebarangkalian) anggaran ko-varian
$Q$	Ko-varian proses yang mempengaruhi ralat disebabkan proses
$R$	Ko-varian pengukuran yang mempengaruhi bunyi dari pengukuran
$rand$	Nombor rawak
$\omega$	Berat inersia
$X_{best}(t)$	Nilai layak yang terbaik dari setiap lelaran
$X_{true}$	Penyelesaian terbaik (sehingga kini)
$\hat{x}_{t-1}$	Keadaan anggaran [ada masa $t-1$
$\hat{x}_{t t-1}$	Jangkaan (keutamaan) anggaran keadaan
$\hat{x}_t$	Terkini (kebarangkalian) anggaran keadaan
$\psi$	Set komponen yang telah dipasang

## SENARAI SINGKATAN

ACO	<i>Ant Colony Optimization</i> Pengoptimuman koloni semut
AMSKF	<i>Angle Modulated Simulated Kalman Filter</i> Penapis Kalman diselakukan dimodulasi sudut
ASP	<i>Assembly Sequence Planning</i> Perancangan jujukan pemasangan
AUTOPASS	<i>Automated Parts Assembly System</i> Sistem pemasangan bahagian automatik
BGSA	<i>Binary Gravitational Search Algorithm</i> Algoritma carian graviti binari
BPSO	<i>Binary Particle Swarm Optimization</i> Pengoptimuman kerumunan zarah binari
BSKF	<i>Binary Simulated kalman Filter</i> Penapis Kalman diselakukan binari
CAD	<i>Computer-Aided Design</i> Sistem reka bantu komputer
CEC	<i>Congress on Evolutionary Computation</i> Kongres untuk pengiraan evolusi
COP	<i>Combinatorial optimization problem</i> Masalah pengoptimuman gabungan
CT	<i>Coefficient Table</i> Jadual kerangka
DESKF	<i>Distance-Evaluated Simulated Kalman Filter</i> Penapis Kalman diselakukan dinilai jarak
GA	<i>Genetic Algorithm</i> Algoritma genetik
GDP	<i>Geometric design processor</i> Pemproses rekaan geometri
GSA	<i>Gravitational Search Algorithm</i> Algoritma carian graviti
GSAA	<i>Combination GA and SA</i> Kombinasi GA dan SA

GSACO	<i>Combination GA, SA and ACO</i> Kombinasi GA, SA dan ACO
IA	<i>Immune Algorithm</i> Algoritma imun
MA	<i>Memetic Algorithm</i> Algoritma pemikiran
MSGSA	<i>Multi-State Gravitational Search Algorithm</i> Algoritma carian graviti berbilang keadaan
MSGSAER	<i>Multi-State Gravitational Search Algorithm with an Embedded Rule</i> Algoritma carian graviti berbilang keadaan dengan peraturan tertanam
MSPSO	<i>Multi-State Particle Swarm Optimization</i> Pengoptimuman kerumunan zarah berbilang keadaan
MSPSOER	<i>Multi-State Particle Swarm Optimization with an Embedded Rule</i> Pengoptimuman kerumunan zarah berbilang keadaan Dengan peraturan tertanam
PADL	<i>Part and assembly description language</i> Bahasa penerangan bahagian dan perhimpunan
PD	<i>Precedence Diagram</i> Rajah utama
PM	<i>Precedence matrix</i> Matrik utama
PSO	<i>Particle Swarm Optimization</i> Pengoptimuman kerumunan zarah
PSOSA	<i>Combination PSO and SA</i> Kombinasi PSO dan SA
SA	<i>Simulated Annealing</i> penyepuhlindungan diselakukan
SKF	<i>Simulated Kalman Filter</i> Penapis Kalman diselakukan
TSP	<i>Travelling Salesman Problem</i> Masalah jurujual mengembara



## **BAB 1**

### **Pengenalan**

#### **1.1 Latar Belakang Kajian**

Sebagai persediaan dalam mendepani persaingan global yang mencabar pada hari ini, pengeluar dan pengilang terlebih dahulu telah mengambil langkah bijak memilih untuk mengurangkan kos produk mereka secara berperingkat-peringkat sejak beberapa tahun yang lalu. Ada banyak cara yang boleh diaplikasikan untuk mencapai matlamat mengurangkan kos sesuatu produk. Antaranya, kaedah perancangan pemasangan iaitu cara yang terbukti mampu mengurangkan masa pemasangan sesuatu produk, seterusnya memberi kesan kepada pengurangan kos pemasangan produk dan pengurangan keseluruhan kos produk.

Proses pemasangan dilihat sebagai prosedur yang penting dalam industri pemasangan. Hampir 50% daripada keseluruhan masa pengeluaran sesuatu produk adalah masa untuk proses pemasangan. Dan, 20% daripada keseluruhan kos produk yang dihasilkan adalah kos untuk proses pemasangan. Perancangan pemasangan sesuatu produk daripada beberapa komponen dapat didefinisikan sebagai pencarian jalan penyelesaian terbaik bagi pemasangan dengan mengambil kira semua konfigurasi yang mungkin bagi satu set komponen yang terlibat. Keadaan permulaannya merupakan konfigurasi di mana semua komponen terpisah antara satu sama lain. Manakala, keadaan penamat atau matlamatnya adalah semua komponen dipasang dengan betul untuk membentuk produk yang diinginkan. Gerakan mengubah satu keadaan menjadi keadaan yang seterusnya adalah sesuai dengan operasi pemasangan yang pastinya perlu menukar kedudukan relatif sekurang-kurangnya satu kali. Kebiasaannya, banyak set jujukan

pemasangan berlainan akan didapati dari keadaan awal hingga keadaan penamat perancangan pemasangan.

Jujukan kepada operasi padanan yang dilakukan untuk memasang satu set komponen seringkali tergendala disebabkan keadaan geometri dan mekanikal komponen-komponen tersebut, konfigurasi pemasangan, serta kestabilan sub-pemasangan yang dihasilkan. Keputusan yang melibatkan teknologi pembuatan pemasangan alternatif, peralatan untuk rekabentuk sistem pembuatan pemasangan, dan kaedah untuk pelaksanaan sistem pemasangan adalah cabaran utama yang kini dihadapi para jurutera pengeluaran. Maka, lebih banyak pendekatan sistematik yang diperlukan untuk analisis, rekabentuk dan perancangan sistem pemasangan bagi meningkatkan prestasi pemasangan dan memastikan pelaksanaan pelan untuk mencapai kos yang efektif berjaya.

Jujukan pemasangan satu set komponen akan menentukan kelajuan dan ketepatan pemasangan produk, seterusnya mempengaruhi prestasi produktiviti. Pengoptimuman pemasangan dalam peringkat perancangan pengeluaran berkaitan dengan penentuan jujukan pemasangan optimum dan penentuan lokasi optimum bagi setiap sumber. Beberapa penyelidik telah mendedikasikan kerja mereka berkenaan beberapa isu penting berkaitan analisis kejuruteraan yang dijalankan serentak pada perancangan jujukan pemasangan (*Assembly Sequence Planning - ASP*). Isu-isu ini merangkumi perkara berkaitan produk-produk yang akan dipasang, penjanaan pelan jujukan pemasangan dan penentuan kekangan utama dalam pemasangan, perwakilan pelan jujukan pemasangan yang dihasilkan dan pemilihan ASP yang optimum.

Dalam industri pengeluaran yang sebenar, menyelesaikan masalah ASP adalah perkara yang sangat penting kerana ia akan menentukan banyak aspek pemasangan iaitu perubahan peralatan, rekabentuk lekapan dan kebebasan pemasangan. Secara praktikal, tenaga kerja manual, automasi tetap, dan automasi fleksibel sering digabungkan dalam sistem pembuatan moden untuk menjimatkan kos dan memenuhi permintaan pasaran.

## **1.2 Pernyataan Masalah**

Masalah pengoptimuman gabungan khas memerlukan interaksi antara jurutera atau dikenali sebagai pakar sistem interaktif dengan maklumat rekabentuk produk untuk mencari jujukan pemasangan optimum dan munasabah. Walau bagaimanapun,

kekurangan maklumat sokongan dalam model produk yang disebabkan oleh banyak maklumat berkaitan rekabentuk pemasangan tidak dapat diterangkan dan disimpan, sukar bagi para jurutera mencari jujukan pemasangan yang optimum dalam sebilangan besar jujukan pemasangan yang mungkin. Tambahan pula, amat sukar untuk merumuskan jujukan pemasangan dengan sejumlah besar jujukan pemasangan yang mungkin tanpa sebarang kesilapan. Justeru, banyak kajian terhadap sistem ASP berdasarkan kemampuan penalaran geometri dan automatik penuh telah giat dijalankan.

Banyak algoritma metaheuristik telah digunakan untuk masalah ASP seperti algoritma carian graviti (*Gravitational Search Algorithm - GSA*) dan pengoptimuman kerumunan zarah (*Particle Swarm Optimization - PSO*). Walau bagaimanapun, kelemahan yang ada pada algoritma-algoritma tersebut seperti kadar penumpuan yang rendah dan kadar pengiraan yang terhad telah membuka ruang untuk penambahbaikan untuk dilakukan oleh algoritma baru.

Algoritma metaheuristik baru bernama penapis Kalman diselakukan (*Simulated Kalman Filter - SKF*) yang mempunyai keupayaan meramal, mengukur dan menganggar, belum pernah diaplikasikan untuk ASP dalam mana-mana kajian. Motivasi kajian ini adalah untuk mengaplikasikan SKF untuk menyelesaikan masalah ASP bagi mendapatkan penyelesaian optimum yang menampung kekurangan algoritma-algoritma sebelum ini.

### **1.3 Objektif Kajian**

Kajian ini dijalankan dengan panduan objektif seperti yang berikut:

1. Untuk mengaplikasikan algoritma SKF lanjutan bagi menyelesaikan masalah pengoptimuman gabungan diskret iaitu ASP.
2. Untuk menganalisis prestasi dan pencapaian algoritma SKF lanjutan dalam menyelesaikan masalah ASP.

## 1.4 Skop Kajian

Skop kerja ini termasuk yang berikut:

1. Dalam kajian ini, aplikasi algoritma SKF lanjutan digunakan untuk menyelesaikan masalah ASP. Oleh kerana masalah ASP sahaja yang dipertimbangkan dalam kajian ini, maka algoritma SKF lanjutan adalah sesuai digunakan kerana ia direka untuk masalah objektif tunggal, bukan objektif pelbagai.
2. Bilangan komponen yang akan dipasang untuk kajian ini adalah 19. Bilangan komponen ini adalah sama dengan yang digunakan dalam kajian terdahulu apabila menggunakan algoritma metaheuristik bagi masalah ASP.
3. Kajian kes dari variasi masalah ASP yang dipilih untuk kajian ini adalah gabungan matrik utama (*Precedence Matrix* – PM) (Choi et al, 1997), jadual kerangka (*Coefficient Jadual* – CT) (Choi et al, 1997) dan Rajah utama (*Precedence Diagram* – PD) (Choi, Lee & Cho, 2008) . Kajian kes ini adalah sama dengan yang digunakan dalam kajian terdahulu apabila menggunakan algoritma metaheuristik kepada masalah ASP. Kemudian, hasil kajian terdahulu akan dirujuk untuk perbandingan dengan hasil kajian ini.
4. Pelbagai platform boleh digunakan untuk melaksanakan algoritma yang dicadangkan seperti MATLAB, Java, dan Visual Basic. Dalam kajian ini, MATLAB dipilih sebagai platform untuk melaksanakan dan mengesahkan algoritma yang dicadangkan.

## 1.5 Manfaat Kajian

Kajian ini menyumbang dalam menyampaikan gambaran menyeluruh mengenai keputusan pencapaian dalam kajian ini untuk mengukur prestasi SKF binari (*Binary SKF* - BSKF), SKF dimodulasi sudut (*Angle Modulated Simulated Kalman Filter* – AMSKF), dan SKF dinilai jarak (*Distance-Evaluated Simulated Kalman Filter* - DESKF), berbanding dengan kajian-kajian sebelumnya bagi algoritma carian graviti binari (*Binary Gravitational Search Algorithm* - BGSA), algoritma pengoptimuman kerumunan zarah binari (*Binary Particle Swarm Optimization* - BPSO), algoritma carian graviti berbilang keadaan (*Multi-State Gravitational Search Algorithm* - MSGSA), algoritma carian graviti berbilang keadaan dengan peraturan tertanam (*Multi-State Gravitational Search*

*Algorithm with an Embedded Rule - MSGSAER*), algoritma pengoptimuman kerumunan zarah berbilang keadaan (*Multi-State Particle Swarm Optimization - MSPSO*), dan algoritma pengoptimuman sekawan zarah berbilang keadaan dengan peraturan tertanam (*Multi-State Particle Swarm Optimization with an Embedded Rule - MSPSOER*) dalam menyelesaikan masalah ASP.

## **1.6 Ringkasan Tesis**

Tesis ini terdiri daripada 5 bab. Bab 1 bermula dengan pembentangan latar belakang kajian, pernyataan masalah, objektif kajian, skop dan manfaat kajian .

Bab 2 mengandungi penjelasan ASP sebagai masalah pengoptimuman gabungan diskret dalam kajian ini. Ini diikuti dengan penjelasan tentang aplikasi algoritma metaheuristik untuk menyelesaikan ASP.

Bab 3 menjelaskan bagaimana algoritma yang dinamakan BSKF, AMSKF, dan DESKF khusus digunakan untuk menyelesaikan masalah diskret seperti dalam ASP. Seterusnya, metodologi kajian memberikan gambaran bagaimana kajian ini dijalankan.

Bab 4 membentangkan hasil pengoptimuman algoritma; BSKF, AMSKF, dan DESKF dalam menyelesaikan ASP. Kemudian, hasil daripada kajian-kajian sebelumnya yang mengaplikasikan BGSA, BPSO, MSPSO, MSPSOER, MSGSA dan MSGSAER untuk menyelesaikan masalah ASP dirujuk dan dibentangkan. Untuk menganalisis keputusan kajian yang dihasilkan oleh algoritma-algoritma dalam menyelesaikan ASP, hasil terbaik bagi setiap algoritma telah diambil untuk perbandingan.

Bab 5 membincangkan dan menyimpulkan sumbangan hasil kajian. Akhirnya, bab ini membincangkan cadangan mengenai kajian masa hadapan.

## BAB 2

### KAJIAN LITERATUR

#### 2.1 Pengenalan

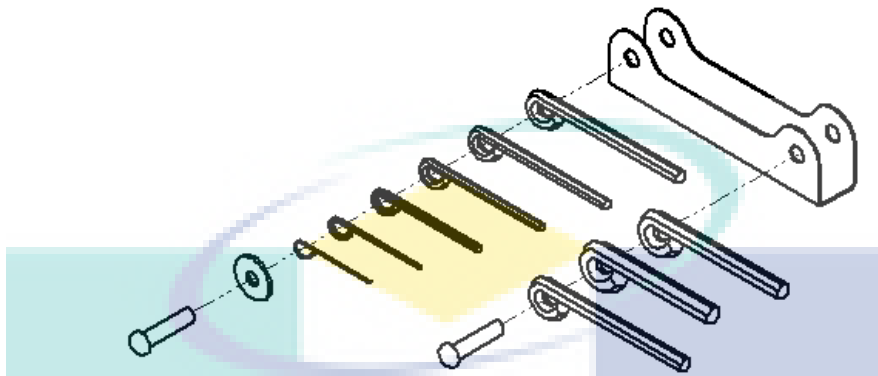
Bab ini memberikan penerangan ringkas masalah utama dalam kajian ini; masalah pengoptimuman gabungan diskret yang dinamakan perancangan jujukan pemasangan atau ASP. Ia diikuti dengan penerangan aplikasi algoritma metaheuristik bagi menyelesaikan masalah ASP.

#### 2.2 Perancangan Jujukan Pemasangan (ASP)

Pemasangan memainkan peranan penting dalam pembuatan produk. Komponen-komponen yang disusun secara berasingan lokasi dan dimensi bagi memenuhi spesifikasi yang dirancang telah dipasang untuk mendapatkan satu konfigurasi yang mencapai fungsi produk atau mekanisma akhir produk tersebut. Kepentingan penjimatan kos pemasangan dalam proses pembuatan telah membawa kepada usaha yang luas untuk meningkatkan kecekapan dan keberkesanan pada kos pemasangan. Rajah 2.1 menunjukkan contoh pandangan bercerai untuk proses pemasangan, di mana komponen-komponen pemasangan diletakkan berasingan. Namun, ia sesuai bersama dan sepadan apabila dipasang.

ASP adalah salah satu bahagian kritikal dalam proses pemasangan. ASP merujuk kepada tugas penganjur iaitu heuristik spesifik mereka dalam mengumpul setiap komponen dan menguasai jujukan pemasangan tertentu. Dalam model pemasangan produk, ASP menentukan jujukan pemasangan komponen untuk memendekkan masa pemasangan atau menjimatkan kos pemasangan. ASP dianggap sebagai masalah pengoptimuman gabungan yang sangat besar dan sangat terhad kerana hampir tidak

mungkin untuk menjana dan menilai semua jujukan yang sesuai untuk mendapatkan jujukan optimum, samada dengan interaksi manusia atau melalui program komputer.



Rajah 2.1 Contoh pandangan meletup bagi proses pemasangan

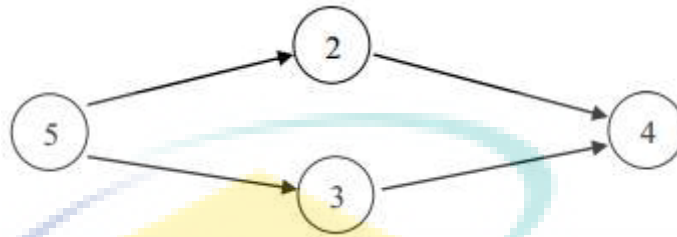
Ruang susunatur meningkat apabila bilangan komponen bertambah dalam ASP. Pertimbangkan situasi dengan enam komponen yang boleh dipasang. Untuk keadaan ini, jumlah jalan penyelesaian jujukan pemasangan yang boleh dibayangkan diberikan oleh  $s = 6!$  yang bersamaan dengan 720 penyelesaian. Apabila kuantiti komponen berkembang kepada tujuh, jumlah penyelesaian jujukan bagi pemasangan yang boleh dibayangkan adalah  $7! = 5,040$ . Walau bagaimanapun, dalam proses pemasangan yang sebenar, terdapat juga kekangan yang perlu dipertimbangkan untuk menjana jujukan pemasangan.

### 2.3 Kekangan ASP

Seperti yang dinyatakan oleh Marian R (2003), terdapat dua jenis kekangan dalam pemasangan, iaitu 'kekangan mutlak' dan 'kekangan pengoptimuman'. Kekangan mutlak merujuk kepada batasan-batasan yang, jika tidak diendahkan, jujukan pemasangan tidak dapat dihasilkan segera. Sebaliknya, kekangan pengoptimuman adalah batasan-batasan yang mempercepatkan kualiti jujukan pemasangan apabila ia tidak diendahkan.

Dalam konteks ASP, kekangan mutlak yang biasanya dipertimbangkan termasuklah keutamaan dan kekangan geometri. Kekangan keutamaan menunjukkan perhubungan komponen pendahuluan dan pengganti untuk proses pemasangan. Kekangan keutamaan perlu diselesaikan. Jika tidak, jujukan pemasangan yang tidak tepat akan dihasilkan. Kekangan keutamaan boleh diwakili dalam Rajah utama (*Precedence Diagram – PD*) atau dalam bentuk matriks. Jadual 2.1 menunjukkan matrik utama

(Precedence Matrix – PM). Dalam matrik ini, apabila bahagian  $a$  mesti dipasang selepas bahagian  $b$ ,  $P(a,b) = 1$ . Jika tidak, matrik akan dibiarkan kosong.



Rajah 2.2 PD bagi pemasangan

Jadual 2.1 PM bagi Rajah 2.2

Komponen $a$	Komponen $b$			
	5	2	3	4
5	0	0	0	0
2	1	0	0	0
3	1	0	0	0
4	1	1	1	0

Dalam masa yang sama, kekangan geometri dalam perhimpunan terikat dengan menggabungkan komponen-komponen tanpa perlanggaran. Apabila memadankan dua komponen, pasti akan ada sekurang-kurangnya satu trajektori yang bebas perlanggaran yang membenarkan pemasangan. Semua jujukan pemasangan yang sah mesti memenuhi kekangan geometri untuk struktur tertentu. Chen dan Liu Chen SF, Liu YJ (2001) menggunakan matrik untuk menggambarkan kekangan geometri antara komponen dalam pemasangan. Bagi setiap pasangan komponen  $(P_a, P_b)$ , matrik akan merekod arah di mana  $P_a$  boleh dipasang dengan  $P_b$  tanpa perlanggaran. Kemudian, satu set arahan pemasangan yang sah, untuk setiap  $(P_a, P_b)$  ditakrifkan, sebagai petunjuk bergerak  $P_a$  dengan  $P_b$ , dirumuskan sebagai PM  $(P_a, P_b)$ . Ia mengira petunjuk bergerak untuk semua pasangan komponen dan menyimpan semua petunjuk bergerak dalam matrik PM.

Manakala, kekangan yang dikelaskan sebagai kekangan pengoptimuman adalah berkaitan dengan objektif pengoptimuman masalah. Kekangan yang diklasifikasikan dalam kategori ini termasuklah kekangan alat pemasangan, kekangan arah pemasangan dan kekangan kestabilan pemasangan.



Wang Y, Liu JH (2010) menggunakan matriks peralatan  $TM = [t_{ab}]_{n \times m}$  untuk mewakili kekangan peralatan. Di mana,  $n$  adalah bilangan komponen dan  $m$  adalah kuantiti peralatan praktikal untuk memasang komponen yang sepadan. Selepas jujukan pemasangan optimum atau yang hampir optimum dijana, peralatan yang sepadan juga disahkan dan pada masa yang sama, bilangan perubahan ( $n_i$ ) peralatan pemasangan boleh diperolehi.

Pada masa yang sama, kekangan arah pemasangan diwakilkan sebagai indeks penalti dalam Wang WP, Tseng HE (2009) Cao PB, Xiao RB (2007), dan Gao L, Qian W, Li X, Wang J (2010). Dalam kajian-kajian tersebut, apabila arah pemasangan berikutnya berbeza dengan arah semasa, satu penalti akan diberikan kepada jujukan pemasangan tertentu itu.

Kekangan kestabilan ditakrifkan oleh Sinanoglu C, Boklu HR (2005) apabila komponen pemasangan mengekalkan kedudukan relatif dan ia tidak memecahkan hubungan semasa operasi pemasangan. Wang Y, Liu JH (2010) mengkategorikan kestabilan yang diperkukuhkan oleh penyambung perhimpunan ke dalam tiga peringkat mengikut kekuatan sambungan; sambungan yang kuat, lemah dan tidak stabil. Klasifikasi ini juga digunakan oleh Yu H, Yu J, Zhang W (2009). Dalam pendekatan ini, sambungan yang kuat akan diberikan indeks '0', indeks sambungan lemah '1' dan indeks sambungan '2' yang tidak stabil.

## 2.4 Objektif ASP

Sejak 10 tahun yang lalu, pelbagai objektif dan teknik pengkomputeran telah digunakan untuk mengoptimumkan masalah ASP. Objektif ASP yang paling popular adalah untuk meminimumkan jumlah perubahan arah pemasangan. Objektif ini dikaji dalam 29 daripada 39 kertas kajian untuk ASP. Untuk mencapai objektif ini, arah pemasangannya adalah di sepanjang tiga paksi utama ( $+x, -x; +y, -y; +z, -z$ ). Apabila arah komponen pemasangan seterusnya berbeza dengan arah semasa, penalti diberikan mengikut magnitud perubahan arah. Jujukan yang optimum adalah jujukan pemasangan yang dikenakan penalti paling minimum disebabkan perubahan arah.

Objektif ASP kedua yang paling popular adalah untuk mengurangkan jumlah pertukaran peralatan. Objektif ini telah dikaji dalam 21 kertas kerja kajian sebelum ini. Dalam proses pemasangan, masa pertukaran peralatan dianggap sebagai salah satu aktiviti yang tidak produktif dan mungkin mengambil masa yang banyak jika tidak diurus dengan baik. Dalam kes ini, apabila proses pemasangan seterusnya memerlukan peralatan yang berbeza dengan proses pemasangan semasa, penalti akan diberikan. Jujukan paling optimum adalah jujukan dengan penalti pertukaran peralatan yang paling minimum.

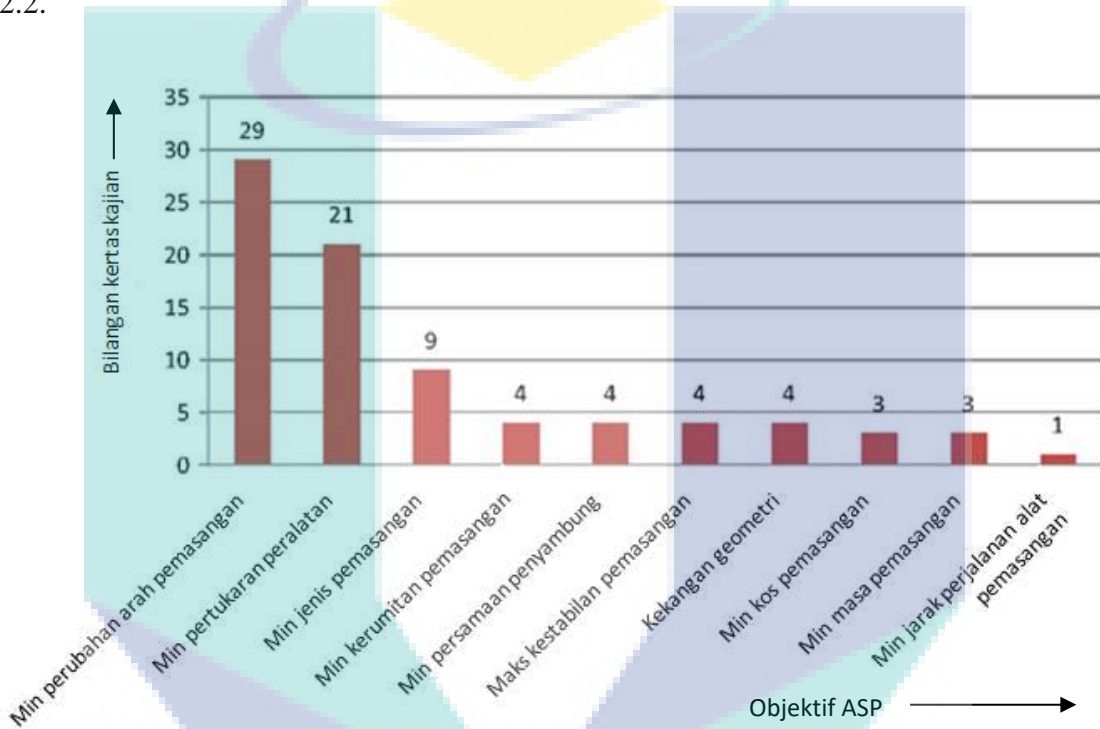
Selain kedua-dua objektif utama ini, objektif ASP untuk meminimumkan pertukaran cara pemasangan telah dikaji dalam 9 kertas kajian. Objektif ini mempertimbangkan perubahan ciri-ciri pemasangan fizikal seperti memadan pemasangan, menyelaras, mengawal, membalikkan dan sebagainya.

Di tempat yang seterusnya, empat objektif ASP berkongsi kedudukan yang sama dengan masing-masing dikaji dalam 4 kertas kajian. Objektif ini adalah untuk meminimumkan kerumitan pemasangan, meminimumkan persamaan penyambung, memaksimumkan kestabilan pemasangan dan meminimumkan penalti kekangan geometri. Dalam ASP, beberapa kajian mendapati bahawa kekangan geometri adalah sekatan wajib. Apabila jujukan pemasangan yang terhasil tidak sepadan dengan kekangan geometri, jujukan pemasangan tidak akan dinilai. Oleh itu, atribut ini tidak dimasukkan sebagai objektif. Walau bagaimanapun, dalam beberapa kajian terhadap ASP seperti Yu H, Yu J, Zhang W (2009), Li JR, Khoo LP, Tor SB (2003), Marian RM, Luong LHS, Abhary K (2006), dan Xing Y, Wang Y, Zhao X (2010), kekangan geometri diisytiharkan sebagai satu objektif ASP. Apabila geometri pemasangan tidak dapat diselesaikan, indeks penalti yang besar akan diberikan kepada fungsi kecergasan. Oleh itu, jujukan yang tidak boleh difahami tidak akan muncul sebagai jujukan pemasangan optimum dalam keputusan akhir.

Rajah 2.3 menunjukkan bahawa tiga objektif ASP yang paling jarang digunakan dalam kertas kajian adalah objektif untuk meminimumkan kos pemasangan, masa pemasangan dan jarak perjalanan peralatan pemasangan. Dalam konteks ASP, matlamat untuk meminimumkan masa pemasangan adalah sesuai untuk digunakan dalam situasi sebenar pemasangan. Sementara itu, objektif untuk meminimumkan jarak perjalanan

peralatan pemasangan adalah berkaitan dengan pemasangan Papan Litar Bercetak yang melibatkan sistem robotik dan penggera.

Perlu diingat bahawa lebih 50% daripada kajian ASP menggunakan teknik pengoptimuman berbilang-objektif yang menggunakan lebih daripada satu objektif dalam kajian mereka. Maka dalam Rajah 2.3, jumlah kekerapan objektif ASP adalah melebihi jumlah kertas kajian ASP. Butiran mengenai maklumat ini boleh didapati dalam Jadual 2.2.



Rajah 2.3 Kekerapan objektif ASP seperti yang diterbitkan dalam kertas kajian

Sumber: Rashid, MFF (2012)

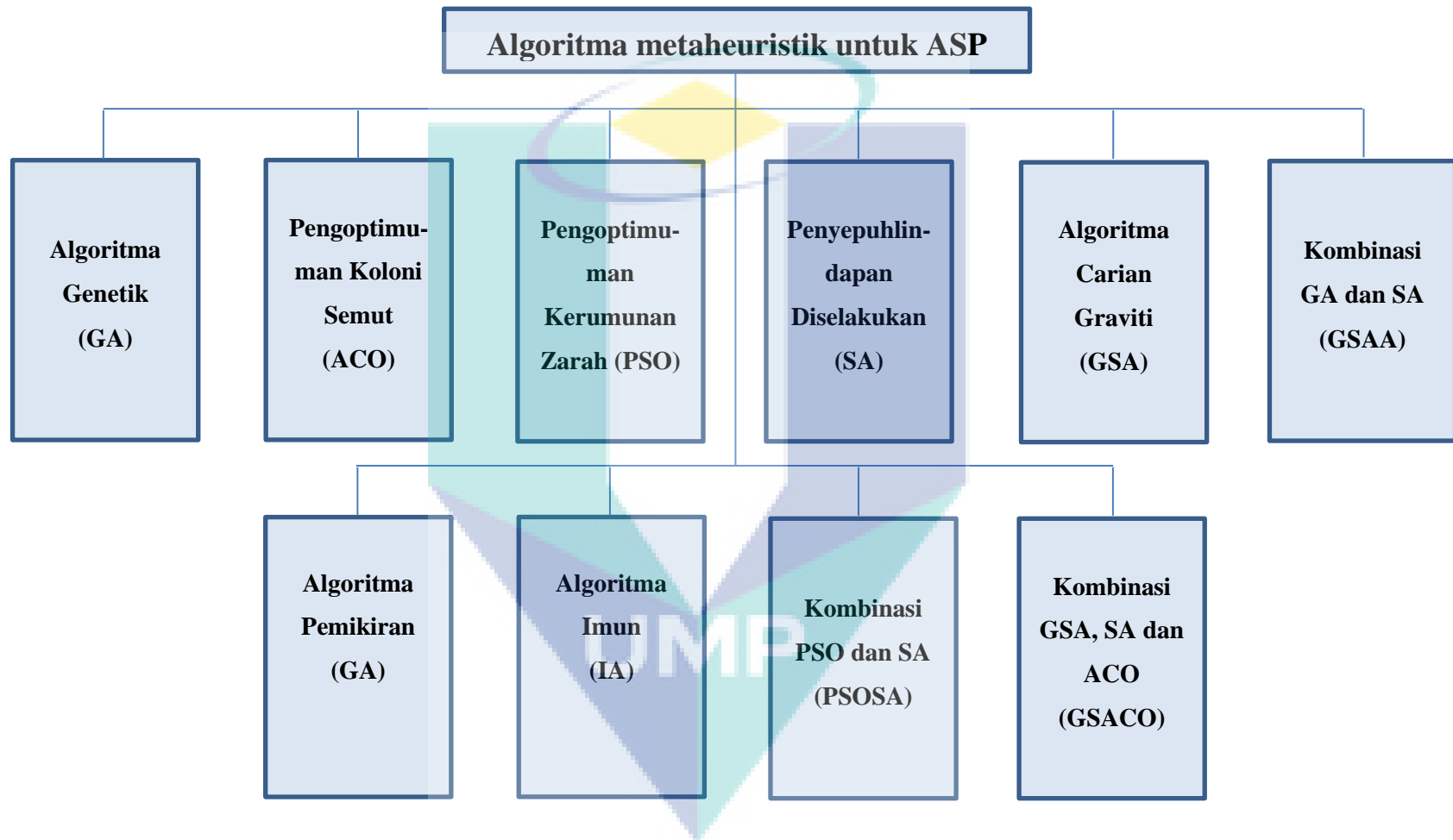
Jadual 2.2 Ringkasan kajian ASP menggunakan kaedah pengkomputeran (2000-2016)

Kaedah	Rujukan	Objektif ASP												
		1	2	3	4	5	6	7	8	9	10	11	12	13
GA	(Chen RS, 2002)	x										x	x	
	(Tseng HE, 2006)	x												x
	(DeLit P, 2001)		x					x		x				
	(Choi YK, 2009)									x				
	(Wang WP, 2009)	x	x		x		x							
	(Chen, 2001)		x											
	(Lazzerini B, 2000)	x	x	x										
	(Guan Q, 2002)	x	x	x										
	(Moon DS, 2007)	x	x											
	(Marian RM, 2006)		x	x						x				
	(Smith SSF, 2001)		x											
	(Smith GC, 2002)		x											
	(Smith SSF, 2004)		x											
	(Tseng HE, 2004)	x	x		x		x							
	(Bai YW, 2005)	x	x											
	(Lu C, 2005)	x	x	x										
	(Udeshi T, 2005)						x							
	(Pan C, 2006)		x											
	(Tseng HE, 2008)	x	x											x
	(Tseng YJ, 2010)	x							x		x			
(Zhou W, 2010)	x	x												
ACO	(Wang JF, 2005)	x												
	(Shuang B, 2008)				x									
PSO	(Lv H, 2010)	x	x	x										
	(Wang Y, 2010)	x	x					x						
	(Yu H, 2009)		x					x	x					
	(Xing Y, 2010)								x					
	(Lv HG, 2010)	x	x	x										
	(Shan H, 2006)												x	
	(Li SX, 2008)												x	
	(Ibrahim I, 2016)												x	
	SA	(Chang CC, 2009)	x	x	x									
GSA	(Cao PB, 2007)										x			
	(Shan H, 2009)										x			
	(Ibrahim I, 2016)										x			

Jadual 2.2 Sambungan.

Kaedah	Rujukan	Objektif ASP												
		1	2	3	4	5	6	7	8	9	10	11	12	13
GSAA	(Hui c, 2009)	x	x	x										
MA	Li JR, 2003)	x	x				x							
	(Su Q, 2009)	x	x				x							
IA	(Chen RS, 2002)	x	x											
PSOSA	(Tseng HE, 2006)	x	x	x										
GSACO	(DeLit P, 2001)				x									

GA (*Genetic Algorithm* - algoritma genetik), ACO (*Ant Colony Optimization* - pengoptimuman koloni semut), PSO (*Particle Swarm Optimization* – pengoptimuman kerumunan zarah), SA (*Simulated Annealing* - penyepuhlindungan diselakukan), GSAA (*combination GA and SA* - kombinasi GA dan SA), GSA (*Gravitational Search Algorithm* - algoritma carian graviti), MA (*Memetic Algorithm* - algoritma pemikiran), IA (*Immune Algorithm* - algoritma imun), PSOSA (*combination PSO and SA* - kombinasi PSO dan SA), GSACO (*combination GA, SA and ACO* - kombinasi GA, SA dan ACO), 1 meminimumkan pertukaran peralatan, 2 meminimumkan perubahan arah pemasangan, 3 meminimumkan pertukaran jenis pemasangan, 4 meminimumkan kerumitan pemasangan, 5 meminimumkan jarak perjalanan alat pemasangan, 6 mengurangkan persamaan penyambung, 7 memaksimumkan kestabilan pemasangan, 8 kekangan geometri, 9 meminimumkan kos pemasangan, 10 meminimumkan masa pemasangan, 11 meminimumkan masa kitaran, 12 memaksimumkan kelancaran beban kerja, 13 meminimumkan bilangan stesen kerja.



Rajah 2.4 Algoritma metaheuristik untuk ASP

## 2.5 ASP untuk Mengurangkan Masa Pemasangan

Untuk meningkatkan produktiviti keseluruhan pengeluaran, pendekatan yang digunakan untuk ASP boleh dikategorikan kepada empat kumpulan, iaitu:

1. **Perwakilan berasaskan grafik.** Dalam perwakilan ini, sumber data berasal dari pengguna atau sistem reka bantu komputer (*Computer-Aided Design - CAD*). Dengan menggunakan pendekatan ini, banyak maklumat analisis pemasangan boleh ditentukan. Mello dan Sanderson (1990) dan Zhang, (1989) mengemukakan kaedah-kaedah perwakilan berasaskan graf berdasarkan kepada DAN / ATAU dan graf yang diarahkan. Lee dan Shin (1990), Moore et al. (2001) dan Zha (2000) mencadangkan kaedah perwakilan berasaskan graf berdasarkan Petri.Lingual representation.
2. **Perwakilan bahasa.** Perwakilan ini menggunakan bahasa khas untuk mewakili sub-pemasangan, komponen dan hubungan antara komponen-komponen. Contoh pendekatan yang melibatkan perwakilan ini adalah bahasa penerangan bahagian dan perhimpunan (*Part and Assembly Description Language - PADL*), sistem pemasangan bahagian automatik (*Automated Parts Assembly System - AUTOPASS*) dan pemproses rekaan geometri (*Geometric design processor - GDP*) (Mello & Sanderson, 1990).
3. **Perwakilan senarai diperintahkan.** Jenis perwakilan ini boleh dikategorikan sebagai senarai peranan tugas yang diperintahkan, vektor binari, sekatan set alat dan sambungan. Garrod (1989) mewakili setiap jujukan pemasangan dalam bentuk set senarai.
4. **Perwakilan berdasarkan meta-heuristik.** Pendekatan ini tidak terhad kepada kaedah berasaskan peraturan (Chakrabarty & Wolter, 1997), pencarian heuristik (Hong & Cho, 1995), rangkaian saraf (Chen, Tai, Deng, & Hsieh, 2008 dan Hsin-Hao et al., 2000), algoritma genetik (Bonnevillle, Perrard, & Henrioud, 1995; Choi et al., 2008; Lit et al., 2010; Lu et al., 2006; Marian et al., 2003; Tseng et al., Dan Zhou et al., 2010), penyepuhlindapan diselakukan (SA) (Milner, Graves, & Whitney, 1994 dan Motavalli & Islam, 1997), Pengoptimuman koloni semut (ACO) (Wang, Liu, Mao, & Fei, 2004), algoritma memetik (Gao, Qian, Li, & Wang, 2009), Pengoptimuman kerumunan zarah (PSO) (Guo, Li, Mileham, &

Owen, 2009; Mukred et al., 2012 dan Tseng et al. dan kaedah hibrid (Hui, Yuan, & Kai-fu, 2008 dan Li et al., 2013).

Pelaksanaan metaheuristik dalam menyelesaikan masalah pengoptimuman diskret, terutamanya dalam masalah ASP, membawa kepada pengurangan ketara dalam masa pengiraan, yang seterusnya memberi jaminan dalam mencari penyelesaian optimum yang tepat (Blum dan Roli, 2013; Talbi, 2009). Walau bagaimanapun, pendekatan ini biasanya mencapai prestasi yang boleh diterima pada kos yang boleh diterima dalam sebilangan besar jujukan pemasangan yang mungkin. Oleh itu, pendekatan ini mempunyai keupayaan untuk mencari penyelesaian yang baik untuk masalah-masalah bersaiz besar.

Untuk membangunkan sistem yang menangani ASP dengan bantuan sistem komputer, terdapat beberapa isu yang perlu diambil kira: jika terdapat sifat sambungan dalam struktur bagi komponen-komponen dan / atau sub-pemasangan yang digunakan dalam sistem pemasangan, berapakah bilangan teoretikal dari sambungan yang berbeza, dan akhirnya pemilihan jujukan pemasangan optimum dan layak daripada pelbagai alternatif.

## **2.6 Model Matematik**

Salah satu objektif utama ASP adalah untuk menjana jujukan pemasangan yang optimum dan layak di mana ia akan mengambil masa minimum untuk memasang, seterusnya mengurangkan kos pemasangan. Faktor yang paling penting dalam mengurangkan masa dan kos pemasangan termasuklah waktu persediaan yang merangkumi masa pemindahan, jumlah kekerapan pertukaran peralatan dan pemilihan ciri-ciri yang sesuai.

Dalam kajian ini, andaian untuk ASP termasuk;

1. Masa persediaan dan masa pemasangan sebenar bagi setiap bahagian dan komponen telah diberikan.
2. Masa pemindahan antara stesen kerja dimasukkan ke dalam masa persediaan.
3. Masa tergendala bagi mesin dan stesen kerja boleh diabaikan.



Masa pemasangan keseluruhan yang dikira dari mesin untuk memasang banyak komponen untuk menjadi produk akhir homogen adalah gabungan dari masa persediaan dan masa pemasangan sebenar. Adalah diandaikan bahawa tanpa mengira jujukan pemasangan, masa pemasangan sebenarnya adalah malar, dan peralatan dan persediaan yang sesuai untuk setiap komponen untuk dipasang diperlukan. Kedua-dua faktor bergantung kepada geometri komponen itu sendiri dan komponen dipasang sehingga siap. Masa persediaan untuk komponen boleh diramalkan menggunakan Persamaan (2.1) dimana  $a$  adalah komponen yang akan dipasang;  $P_{a0}$  ialah masa persediaan dengan komponen  $a$  menjadi komponen pertama dipasang;  $P_{ab}$  adalah penyumbang kepada masa persediaan kerana kehadiran komponen  $b$  apabila memasuki komponen  $a$ ; dan nilai  $q_{ab} = 1$  jika komponen  $b$  telah dipasang pada  $a$ . Manakala,  $q_{ab} = 0$  jika komponen  $b$  gagal dipasang pada  $a$ . Nilai  $a = 1, 2, \dots, c$ , di mana  $c$  merupakan bilangan komponen dalam masalah ASP tersebut.

$$Time_{Setup}(a) = p_{a0} + \sum_{b=1}^c p_{ab} q_{ab} \quad (2.1)$$

Masa pemasangan keseluruhan adalah penjumlahan masa persediaan dan masa pemasangan sebenar. Oleh itu, fungsi objektif untuk meminimumkan masa pemasangan dijelaskan oleh Persamaan (2.2) di mana  $A_a$  ialah masa pemasangan untuk komponen  $a$  dari meja pekali. Pengiraan masa adalah dalam unit masa.

$$\text{Min } Time_{Assembly} = \sum_{a=1}^c (Time_{Setup}(a) + A_a) \quad (2.2)$$

Untuk mendapatkan jujukan pemasangan yang munasabah, semua jujukan pemasangan yang dihasilkan mesti mematuhi semua kekangan yang terdahulu. Salah satu kekangan reka bentuk perhimpunan adalah hubungan keutamaan antara komponen. Dalam kajian ini, pemasangan produk dengan lima komponen diwakili dalam graf yang diarahkan, seperti ditunjukkan dalam Rajah 2.2. Secara umum, data input yang diperlukan dalam proses pemasangan mudah diekstrak sama ada dari CAD atau analisis pembongkaran.

PM seperti yang ditunjukkan dalam Jadual 2.4 digunakan untuk menunjukkan hubungan antara komponen dalam pemasangan yang menggunakan kekangan sebelum ini. Hubungan ini termasuk sifat sambungan (iaitu, komponen bebas atau dipasang) dan

kehadiran perhubungan relatif antara dua komponen. Untuk tujuan ini, Jadual 2.1 boleh dibina mengikut kekangan sebelumnya antara komponen  $a$  dan  $b$ . Jika komponen mesti dipasang selepas komponen  $b$ ,  $PM(P_a, P_b) = 1$ ; sebaliknya  $PM(P_a, P_b) = \emptyset$ , di mana  $(P_a, P_b)$  adalah sepasang komponen dengan maklumat geometri di mana  $P_a$  mesti dipasang tanpa mengganggu  $P_b$ .

Untuk menentukan pasangan mana yang boleh dipadankan, kekangan keutamaan untuk sesuatu produk harus diterangkan menggunakan PM. Dengan mengandaikan bahawa  $\psi$  ialah set komponen yang telah dipasang sebelum komponen  $a$ , dan PM adalah jujukan perakitan yang layak  $FAS(P_a, P_b)$  dengan kekangan, maka:

$$FAS(P_a, P_b) = \cup PM(P_a, P_b), P_b \in \psi \quad (2.3)$$

Persamaan (2.3) menjelaskan bahawa komponen boleh dipasang jika  $FAS(P_a, P_b)$  adalah satu set kosong kerana semua komponen yang mesti dipasang sebelum komponen telah dipasang. Sebagai contoh, komponen 4 dalam Rajah 2.2 hanya boleh dipasang jika komponen 5, 2 dan 3 telah dipasang. Oleh itu, jujukan pemasangan 5-2-3-4 dapat dilaksanakan. Kajian ini menunjukkan bahawa komponen 5 mestilah komponen pertama yang dipasang kerana  $P_5$  adalah satu set kosong untuk semua  $P_b$ . Seterusnya, sama ada komponen 2 atau 3 boleh dipilih sebagai komponen kedua yang akan dipasang kerana jika  $FAS(P_2, P_b) = (P_b = 5)$ . Jika komponen 2 dipilih, komponen yang mesti dipasang sebelum komponen 2 hanya boleh menjadi komponen 5. Komponen 3 kemudiannya boleh dipilih sebagai komponen ketiga yang akan dipasang kerana jika  $FAS(P_3, P_b) = (P_b = 5)$ , komponen yang mesti dipasang sebelum komponen 3 telah dipasang (iaitu komponen 5). Akhir sekali, komponen 4 boleh dipilih sebagai komponen keempat yang akan dipasang kerana jika  $FAS(P_4, P_b) = (P_b = 3)$ , komponen yang mesti dipasang sebelum komponen 4 telah dipasang (iaitu komponen 5, 2, dan 3).

Banyak algoritma metaheuristik telah digunakan untuk menyelesaikan masalah ASP. Jadual 2.2 menunjukkan algoritma yang telah digunakan dalam kajian yang lalu bagi tujuan meminimumkan masa pemasangan adalah PSO dan GSA.

## 2.7 Pengoptimuman Kawanan Zarah (PSO)

Pengoptimuman kawanan zarah (PSO) adalah algoritma stokastik berasaskan populasi yang digunakan untuk mencari penyelesaian yang optimum (atau hampir optimum) untuk masalah pengoptimuman. Algoritma sedemikian yang mudah digunakan (hanya beberapa baris kod perlu ditulis bagi algoritma ini) boleh dinamakan sebagai algoritma yang luar biasa dan pantas untuk masalah pengoptimuman yang berbeza.

Algoritma PSO adalah algoritma pengoptimuman yang dicadangkan oleh Kennedy dan Eberhart (1995). Ia meniru tingkahlaku kawanan seperti yang ditunjukkan oleh kawanan burung dan kawanan ikan untuk mencari penyelesaian optimum yang tertakluk kepada fungsi objektif.

Menurut Mukred, J.A.A. (2012), dalam algoritma PSO, penyelesaian yang optimum atau baik didapati dengan meniru tingkah laku sosial burung berkumpul. Algoritma PSO terdiri daripada sekumpulan individu bernama zarah yang memberikan kemungkinan penyelesaian kepada masalah pengoptimuman menggunakan kedudukan mereka. Kumpulan dapat mencapai penyelesaian secara efektif dengan menggunakan maklumat umum kumpulan dan maklumat yang dimiliki oleh zarah itu sendiri, yang mana setiap zarah berkongsi kedudukannya sekarang untuk zarah-zarah yang berdekatan. Menggunakan maklumat ini, setiap zarah membandingkan kedudukan semasa dengan kedudukan terbaik yang dijumpai oleh jirannya setakat ini.

Pengoptimuman merangkumi masalah meminimumkan dan memaksimumkan. Masalah pemaksimuman boleh ditukar menjadi masalah meminimumkan dengan mengambil negatif fungsi objektif, dan sebaliknya. Rajah 2.5 menggambarkan pseudokod algoritma PSO. Pertimbangkan masalah pengoptimuman berikut: terdapat  $I$ -zarah yang terbang di dalam ruang carian  $D$ -dimensi, di mana kedudukannya,  $s_i(d)$  ( $i = 1, 2, \dots, I$ ;  $d = 1, 2, \dots, D$ ), mewakili penyelesaian yang mungkin dan  $d$  mewakili nombor dimensi. Pertama, semua zarah secara rawak diposisikan dalam ruang carian, dan kemudian ditetapkan dengan halaju  $v_i(k, d) = 0$ , di mana  $k$  mewakili bilangan lelaran. Seterusnya, kecergasan objektif  $F_i(k)$ , bagi setiap zarah dinilai dengan mengira fungsi objektif berkenaan dengan  $S_i(k)$ . Setiap kedudukan terbaik zarah adalah  $pbest_i(k)$  kemudian diisytiharkan ke kedudukan semasa. Yang terbaik global di kalangan semua  $pbest_i(k)$

dipanggil  $gbest(k)$ , dipilih sebagai kedudukan terbaik. Untuk masalah meminimumkan,  $gbest(k)$  diberikan seperti dalam Persamaan (2.4). Sementara itu, untuk masalah memaksimumkan,  $gbest(k)$  diberikan seperti dalam Persamaan (2.5) Dalam kedua-dua persamaan ini,  $S$  adalah kumpulan zarah. Selepas itu, algoritma dijalankan sehingga kriteria berhenti dipenuhi, ralat dicapai atau bilangan maksimum lelaran digunakan.

$$gbest = \{pbest_i \in S | f(pbest_i) = \min f(\forall pbest_i \in S)\} \quad (2.4)$$

---

```

1: prosedur PSO
2:   Permulaan zarah dengan posisi dan halaju rawak
3:   Setkan zarah  $pbest$  kepada posisi terkini
4:   Kira kelayakan zarah dan set  $gbest$ 
5:   for generasi  $T$  do
6:     Kemaskini halaju zarah
7:     Kemaskini posisi zarah
8:     Kira semula kelayakan zarah
9:     Kemaskini  $pbest$  and  $gbest$ .
10:  end for
11: end procedure PSO

```

---

Rajah 2.5. Algoritma PSO

$$gbest = \{pbest_i \in S | f(pbest_i) = \max f(\forall pbest_i \in S)\} \quad (2.5)$$

Setiap lelaran mengemaskini halaju dan kedudukan zarah menggunakan Persamaan (2.6) dan Persamaan (2.7), masing-masing, di mana  $c_1$  dan  $c_2$  masing-masing adalah pekali kognitif dan sosial.  $r_1$  dan  $r_2$  adalah nombor rawak yang ditaburkan secara seragam antara 0 dan 1, dan  $\omega$  dipanggil berat inerti, yang digunakan untuk mengawal kesan halaju sebelumnya kepada halaju semasa setiap zarah. Selepas mengemaskini halaju dan kedudukan,  $F_i^{\rightarrow}(k)$  bagi setiap zarah dikira semula.  $pbest_i(k)$  kemudian dikemas kini dengan yang lebih optimum, diperoleh sama ada dari kedudukan baru zarah ke- $i$  atau  $pbest_i(k)$ .  $gbest(k)$  juga dikemas kini oleh  $pbest_i(k)$  yang paling optimum dari semua zarah, seperti yang ditunjukkan dalam Persamaan (2.4) dan Persamaan (2.5). Akhirnya, penyelesaian terbaik masalah yang diwakili oleh  $gbest(k)$  dihasilkan apabila keadaan berhenti dipenuhi.

$$v_i(k+1,d) = \omega v_i(k,d) + c_1 r_1 (pbest_i)(k,d) - s_i(k,d) + c_2 r_2 (gbest(k,d) - s_i(k,d)) \quad (2.6)$$

$$s_i(k+1,d) = s_i(k,d) + v_i(k+1,d) \quad (2.7)$$

## 2.8 Algoritma Carian Graviti (GSA)

Pada tahun 2009, metaheuristik berasaskan populasi stokastik yang dikenali sebagai algoritma carian graviti (GSA) telah diperkenalkan (Rashedi, Nezamabadi-pour, & Saryazdi, 2009) berdasarkan hukum graviti universal Newton. Peraturan ini menunjukkan bahawa semua objek menarik satu sama lain menggunakan daya graviti. Ciri pertama dari gaya graviti antara dua objek adalah sama dengan lajur produk. Objek yang lebih besar akan menarik satu sama lain dengan daya graviti yang lebih besar. Ciri kedua daya graviti adalah ia berkadar terus berbanding dengan jarak antara kedua-dua objek. Versi asal GSA pada asalnya dimodelkan untuk menangani masalah dalam ruang yang memaparkan vektor bernilai sebenar.

Rajah 2.6 menggambarkan pseudokod GSA. GSA bermula dengan satu set ejen, yang secara rawak diposisikan dalam ruang carian. Biarkan sistem dengan  $I$ -ejen (jisim), kedudukan ejen yang mewakili penyelesaian yang mungkin untuk masalah boleh ditakrifkan sebagai  $s_i(d)$  ( $i = 1, 2, \dots, I; d = 1, 2, \dots, D$ ), di mana  $s_i(d)$  ialah kedudukan ejen ke- $i$  dalam dimensi ke- $d$ . Semua ejen kemudian diberikan dengan halaju rawak  $v_i(k,d)$ , di mana  $k$  mewakili bilangan lelaran. Seterusnya, kecergasan objektif  $F^i(k)$  bagi setiap ejen dinilai dengan mengira fungsi objektif berkenaan dengan  $s_i(k+d)$ , di mana  $F^i(k)$  mewakili nilai kecergasan agen  $i$  di  $k$ . Pemalar graviti  $G(k)$  kemudian dikemaskini berdasarkan Persamaan (2.8). Pemalar graviti adalah fungsi penurunan masa di mana ia ditetapkan kepada  $G_0$  pada permulaan dan secara eksponen menurun ke arah sifar pada lelaran terakhir untuk mengawal ketepatan carian

---

1: **prosedur GSA**

- 2: Permulaan ejen dengan posisi dan halaju rawak
  - 3: Setkan ejen kepada posisi terkini
  - 4: Kira kelayakan ejen dan set  $G$ , ejen terbaik dan ejen terburuk
  - 5: **for** generasi  $T$  **do**
  - 6:     Menilai jisim
  - 7:     Menilai daya bagi jisim
  - 8:     Menilai pecutan bagi jisim
  - 9:     Kemaskini halaju ejen
  - 10:    Kemaskini posisi ejen
  - 11:    Kira semula kelayakan ejen
  - 12:    Kemaskini  $G$ , ejen terbaik dan terburuk
  - 13: **end for**
  - 11: **end procedure GSA**
- 

Rajah 2.6. Algoritma GSA

$$G(k) = G_0 e^{-\beta \frac{k}{K}} \quad (2.8)$$

di mana  $K$  adalah bilangan lelaran maksimum,  $G_0$  dan  $\beta$  adalah nilai malar.

Seterusnya, terbaik  $best(k)$  dan terburuk  $worst(k)$  dikira. Untuk masalah meminimumkan, definisi  $best(k)$  dan  $worst(k)$  diberikan seperti dalam Persamaan (2.9) dan Persamaan (2.10).

$$best(k) = \min_{j \in \{1, \dots, J\}} \vec{F}_j(k) \quad (2.9)$$

$$worst(k) = \max_{j \in \{1, \dots, J\}} \vec{F}_j(k) \quad (2.10)$$

Untuk masalah memaksimumkan, definisi  $best(k)$  dan  $worst(k)$  ditukar seperti dalam Persamaan (2.11) dan Persamaan (2.12):

$$best(k) = \max_{j \in \{1, \dots, J\}} \vec{F}_j(k) \quad (2.11)$$

$$worst(k) = \min_{j \in \{1, \dots, J\}} \vec{F}_j(k) \quad (2.12)$$

Jisim graviti dan inersia kemudian dikemaskini menggunakan Persamaan (2.13) dan Persamaan (2.14):

$$m_i(k) = \frac{\bar{F}_i(k) - \text{worst}(k)}{\text{best}(k) - \text{worst}(k)} \quad (2.13)$$

$$M_i(k) = \frac{m_i(k)}{\sum_{j=1}^I m_j(k)} \quad (2.14)$$

$M_{ii}(k)$  adalah jisim inersia ejen ke- $i$ . Pecutan  $\alpha$  yang berkaitan dengan jisim  $i$  dalam  $k$  dan dimensi ke- $d$  dikira seperti Persamaan (2.15):

$$\alpha_i(k) = \frac{F_i(k,d)}{M_{ii}(k)} \quad (2.15)$$

Jumlah daya  $F_i(k,d)$  dan daya yang bertindak pada jisim  $i$  dari jisim  $j$   $F_{ij}(k,d)$  dikira seperti Persamaan (2.16) dan Persamaan (2.17):

$$F_i(k,d) = \sum_{j=1, j \neq i}^I \text{rand}_j F_{ij}(k,d) \quad (2.16)$$

$$F_{ij}(k,d) = G(k) \frac{M_{aj}(k)}{R_{ij}(k) + \varepsilon} (s_j(k,d) - s_i(k,d)) \quad (2.17)$$

di mana  $M_{aj}$  adalah jisim graviti aktif yang berkaitan dengan ejen  $j$ ,  $\varepsilon$  adalah pemalar kecil,  $R_{ij}$  ialah jarak Euclidean antara ejen  $i$  dan  $j$ , dan  $\text{rand}_j$  adalah bilangan rawak yang diedarkan secara seragam [0, 1]. Seterusnya, algoritma berjalan sehingga kriteria penghentian dipenuhi, sama ada jumlah kesilapan yang boleh diterima yang telah ditetapkan sebelumnya atau bilangan maksimum lelaran yang digunakan.

$$v_i(k+1,d) = \text{rand}_i \times v_i(k,d) + \alpha_i(k) \quad (2.18)$$

$$s_i(k+1,d) = s_i(k,d) + v_i(k+1,d) \quad (2.19)$$

Setiap lelaran mengemaskini halaju dan kedudukan ejen menggunakan Persamaan (2.18) dan Persamaan (2.19), di mana  $rand_i$  adalah nombor rawak yang ditaburkan secara seragam  $[0, 1]$ . Selepas mengemaskini halaju dan kedudukan,  $F^{-i}(k)$  bagi setiap ejen dikira semula. Pertimbangkan masalah meminimumkan, yang terbaik dari populasi kemudian dikemaskini berdasarkan yang paling minimum, samada diperolehi dari kedudukan terkini ejen ke- $i$  atau yang terbaik sedia ada. Yang terburuk dalam populasi juga dikemaskini berdasarkan yang lebih maksimum, samada diperolehi dari kedudukan terkini ejen ke- $i$  atau yang terburuk sedia ada. Pertimbangkan masalah memaksimumkan, yang terbaik dari populasi kemudian dikemaskini berdasarkan yang paling maksimum, samada diperolehi dari kedudukan terkini ejen ke- $i$  atau yang terbaik sedia ada. Yang terburuk dalam populasi juga dikemaskini berdasarkan yang lebih minimum, samada diperolehi dari kedudukan terkini ejen ke- $i$  atau yang terburuk sedia ada. Akhirnya, penyelesaian terbaik masalah yang diwakili oleh yang terbaik dihasilkan apabila keadaan berhenti dipenuhi.

## 2.9 Algoritma Penapis Kalman

Penapis Kalman berguna untuk menjejaki pelbagai objek bergerak. Ia pada mulanya diperkenalkan oleh Kalman (1960) dengan menggabungkan pengukuran bising dan ramalan keadaan untuk mencapai anggaran keadaan semasa yang sebenar. Penggunaan tipikal bagi penapis Kalman termasuklah menyunting data bising dan memberikan anggaran kepentingan parameter. Dalam algoritma penapis Kalman, definisi bagi terma yang diwakili diberikan seperti berikut:

- **Vektor Keadaan:** Mengandungi parameter atau pemboleh ubah yang dikehendaki, seperti lokasi objek, halaju, voltan, dan suhu. Dalam sudut pandangan algoritma pengoptimuman, keadaan boleh dirujuk sebagai kedudukan ejen dalam ruang carian.
- **Vektor pengukuran:** Vektor yang mengandungi input pengukuran seperti kedudukan dan bacaan halaju dari sensor.
- **Matriks peralihan keadaan:** Matriks yang digunakan untuk mengekstrapolasi ciri-ciri deterministik vektor negeri dari masa  $t-1$  hingga  $t$ .
- **Matriks kovarian parameter:** Mengandungi anggaran ketidakpastian dan korelasi antara ketidaktentuan komponen vektor negeri.



- **Matriks kovarian proses:** Matrik yang menerangkan bunyi atau hingar proses. Contohnya, katakan penapis Kalman digunakan untuk mengesan kedudukan dan halaju dalam paksi  $x$  untuk sebuah kereta. Bunyi proses mungkin perkara seperti angin dan lebam di jalan raya.
- **Matriks kovarian pengukuran:** Menghuraikan bunyi pengukuran, yang tidak tepat yang diperkenalkan oleh sensor itu sendiri seperti turun naik bekalan kuasa.
- **Keuntungan Kalman:** Menentukan pemberat pengukuran terkini dan anggaran dari lalaran sebelumnya. Nilai ini dikira oleh penapis.
- **Anggaran *Priori*:** Anggaran kedudukan dan anggaran ralat kovarian tanpa mempertimbangkan ukuran yang ada.
- **Anggaran *Posteriori*:** Anggaran anggaran keadaan dan kovarians ralat dengan memasukkan pengukuran yang ada.
- **Kemas kini masa:** Bertanggungjawab untuk mengunjurkan (dalam masa) keadaan semasa anggaran ralat kovarians untuk memperolehi anggaran *priori* untuk langkah seterusnya.
- **Kemaskini pengukuran:** Bertanggungjawab untuk maklum balas dengan memasukkan pengukuran proses ke dalam anggaran *priori* untuk mendapatkan anggaran *posteriori* lanjutan.

‘Keadaan’ boleh dirujuk kepada kuantiti yang boleh diukur, seperti lokasi objek, halaju, voltan, dan suhu. Ia menganggap bahawa keadaan sistem pada masa  $t$  berubah dari keadaan terdahulu iaitu pada  $t-1$  mengikut Persamaan (2.6).

$$\mathbf{x}_t = A\mathbf{x}_{t-1} + B\mathbf{u}_t + w_t \quad (2.6)$$

di mana  $x_t$  ialah vektor keadaan yang mengandungi parameter yang dianggarkan pada masa  $t$ , iaitu vektor input kawalan,  $A$  adalah matriks peralihan keadaan yang mengaitkan keadaan pada masa  $t-1$  kepada keadaan pada masa  $t$ ,  $B$  ialah matriks menghubungkan input kawalan pilihan kepada keadaan, dan  $w_t$  adalah vektor yang mengandungi bunyi proses.

Pengukuran sistem juga boleh dilakukan berdasarkan Persamaan (2.20).

$$\mathbf{z}_t = H\mathbf{x}_t + \mathbf{v}_t \quad (2.20)$$

di mana  $\mathbf{z}_t$  ialah vektor pengukuran pada masa  $t$ ,  $H$  ialah matriks yang mentakrifkan pemetaan dari vektor keadaan kepada vektor pengukuran, dan  $\mathbf{v}_t$  adalah vektor yang mengandungi bunyi pengukuran.

Penapis Kalman adalah algoritma dua peringkat, termasuk kemas kini masa dan kemas kini pengukuran. Persamaan untuk pembaharuan masa, seperti yang ditunjukkan dalam Persamaan (2.21) dan Persamaan (2.22) adalah bertanggungjawab untuk mengunjurkan keadaan semasa dan anggaran ralat kovarian untuk mendapatkan perkiraan *priori* untuk langkah seterusnya.

$$\hat{\mathbf{x}}_{t|t-1} = A\hat{\mathbf{x}}_{t-1} + B\mathbf{u}_t \quad (2.21)$$

$$P_{t|t-1} = AP_{t-1}A^T + Q \quad (2.22)$$

di mana  $\hat{\mathbf{x}}_{t|t-1}$  ialah keadaan yang diramalkan (*priori*),  $\hat{\mathbf{x}}_{t-1}$  ialah taksiran keadaan pada masa  $t-1$ ,  $P_{t|t-1}$  adalah kovarians yang diramalkan (*priori*) ketepatan anggaran,  $P_{t-1}$  ialah anggaran kovarian pada masa  $t-1$ , dan  $Q$  ialah kovarian proses yang mencerminkan ralat daripada proses.

Persamaan kemas kini pengukuran seperti yang ditunjukkan dalam Persamaan (2.23), Persamaan (2.24), dan Persamaan (2.25) bertanggungjawab untuk maklumbalas. Tujuan persamaan ini adalah untuk memasukkan pengukuran baru ke dalam perkiraan *priori* untuk mendapatkan perkiraan *posteriori* lanjutan.

$$K_t = P_{t|t-1}H^T(HP_{t|t-1}H^T + R)^{-1} \quad (2.23)$$

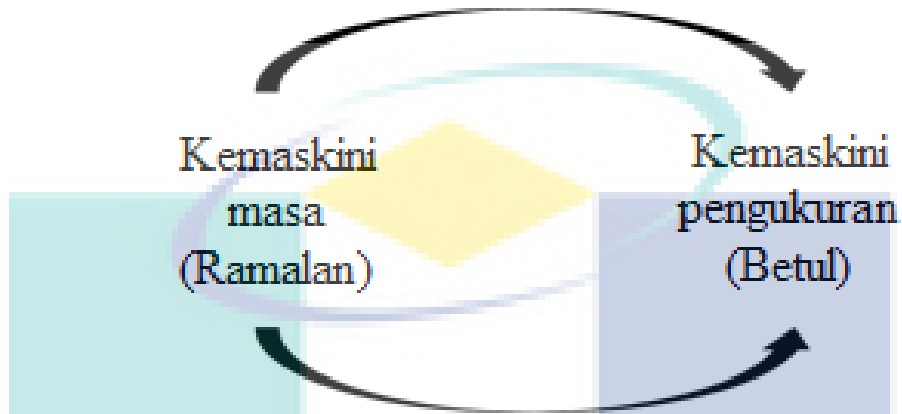
$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t|t-1} + K_t(\mathbf{z}_t - H\hat{\mathbf{x}}_{t|t-1}) \quad (2.24)$$

$$P_t = (I - K_tH)P_{t|t-1} \quad (2.25)$$

di mana  $K_t$  adalah keuntungan Kalman,  $R$  adalah kovarians pengukuran yang mencerminkan bunyi dari pengukuran,  $\hat{\mathbf{x}}_t$  ialah anggaran keadaan (*posteriori*) yang dikemaskini, dan  $P_t$  adalah kovarian anggaran (*posteriori*) yang dikemaskini.

Persamaan pembaharuan masa juga boleh dianggap sebagai persamaan *ramalan*, manakala persamaan pemboleh ubah pengukuran dapat dianggap sebagai persamaan

*pembetul*. Seperti yang ditunjukkan dalam Rajah 2.7, proses ini berulang-ulang dengan sebelumnya perkiraan *posteriori* yang digunakan untuk meramalkan anggaran *priori* baru. Sifat pengulangan ini adalah salah satu ciri yang menarik pada penapis Kalman.



Rajah 2.7 Kitaran penapis Kalman yang berterusan  
Sumber: Welch and Bishop (1995)

## 2.10 Algoritma Penapis Kalman Diselakukan (SKF)

Ibrahim et al. (2015) memperkenalkan algoritma pengoptimuman metaheuristik baru disebut sebagai Penapis Kalman Diselakukan (SKF). Algoritma terkini ini diilhamkan oleh keupayaan anggaran penapis Kalman. Keputusan eksperimen berdasarkan *30 Congress on Evolutionary Computation (CEC) 2014* menunjukkan fungsi uji tanda aras bahawa algoritma tersebut menawarkan prestasi setanding dengan beberapa algoritma metaheuristik yang dibandingkan. Selain itu, kadar penumpuan ejen dalam SKF juga cepat dalam mencari optimum global. Walau bagaimanapun, kepelbagaiannya berkurangan dengan cepat dalam lelaran berturut-turut dan menyekat algoritma untuk menerokai kawasan yang lebih tinggi kebarangkalian berjumpa optimum di ruang carian. Oleh kerana SKF adalah algoritma pengoptimuman baru, sangat sedikit kertas kerja kajian yang dilaporkan untuk SKF.

SKF adalah algoritma metaheuristik berasaskan populasi yang diperkenalkan untuk masalah pengoptimuman yang berterusan. Algoritma SKF ditunjukkan dalam Rajah 2.9. Algoritma ini bermula dengan pengenalan agen-agen  $n$ , di mana kedudukan setiap ejen diwujudkan secara rawak dalam ruang pencarian. Nilai permulaan anggaran

kovarians ralat,  $P(0)$ , nilai hingar proses,  $Q$ , dan nilai hingar pengukuran,  $R$ , yang diperlukan dalam penapisan Kalman, juga ditentukan semasa peringkat permulaan.

Selepas itu, setiap ejen tertakluk kepada penilaian kecergasan. Nilai kecergasan diperiksa dan ejen yang memiliki nilai kecergasan terbaik pada setiap lelaran,  $t$ , dicatatkan sebagai  $X_{best}(t)$ .

Untuk masalah meminimumkan fungsi,

$$X_{best}(t) = \min_{i \in \{1, \dots, n\}} fit_i(X(t)) \quad (2.26)$$

dan untuk masalah memaksimumkan fungsi,

$$X_{best}(t) = \max_{i \in \{1, \dots, n\}} fit_i(X(t)) \quad (2.27)$$

Penyelesaian terbaik dalam SKF dinamakan  $X_{true}$ . Ia dikemaskini hanya jika  $X_{best}(t)$  lebih baik ( $X_{best}(t) < X_{true}$  untuk masalah meminimumkan, atau  $X_{best}(t) > X_{true}$  untuk masalah memaksimumkan) berbanding  $X_{true}$ .

Pengiraan seterusnya pada dasarnya sama dengan prosedur ramalan, pengukuran, dan anggaran dalam Kalman Filter. Dalam peringkat ramalan, persamaan pembaharuan masa berikut dikira:

$$X_i(t|t) = X_i(t) \quad (2.28)$$

$$P(t|t) = P(t) + Q \quad (2.29)$$

di mana  $X_i(t|t)$  dan  $X_i(t)$  adalah keadaan yang diramalkan dan terdahulu, yang mengandungi kedudukan agen dalam ruang carian.  $P(t|t)$  dan  $P(t)$  masing-masing adalah ramalan anggaran kovariant ralat dan anggaran kovariant ralat sebelumnya, dan  $Q$  ialah bunyi proses.

Langkah seterusnya ialah ukuran, yang bertindak sebagai maklum balas kepada proses anggaran. Pengukuran dimodelkan supaya outputnya boleh mengambil sebarang nilai dari anggaran keadaan yang diramalkan,  $X_i(t|t)$ , kepada nilai sebenar,  $X_{true}$ . Pengukuran,  $Z_i(t)$ , setiap ejen individu disimulasikan mengikut Persamaan (2.30).

$$\mathbf{Z}_i(t) = \mathbf{X}_i(t|t) + \sin(rand \times 2\pi) \times |\mathbf{X}_i(t|t) - \mathbf{X}_{true}| \quad (2.30)$$

di mana istilah  $\sin(rand \times 2\pi)$  mewakili bahagian stokastik SKF dan  $rand$  adalah nombor rawak yang ditabur seragam antara 0 hingga 1. Langkah terakhir adalah anggaran. Dalam langkah ini, keuntungan Kalman,  $K(t)$ , dikira seperti berikut:

$$K(t) = \frac{P(t|t)}{P(t|t) + R} \quad (2.31)$$

Kemudian, anggaran keadaan seterusnya,  $\mathbf{X}_i(t+1)$ , dikira berdasarkan Persamaan (2.32) dan ralat kovarians dikemaskini berdasarkan Persamaan (2.33).

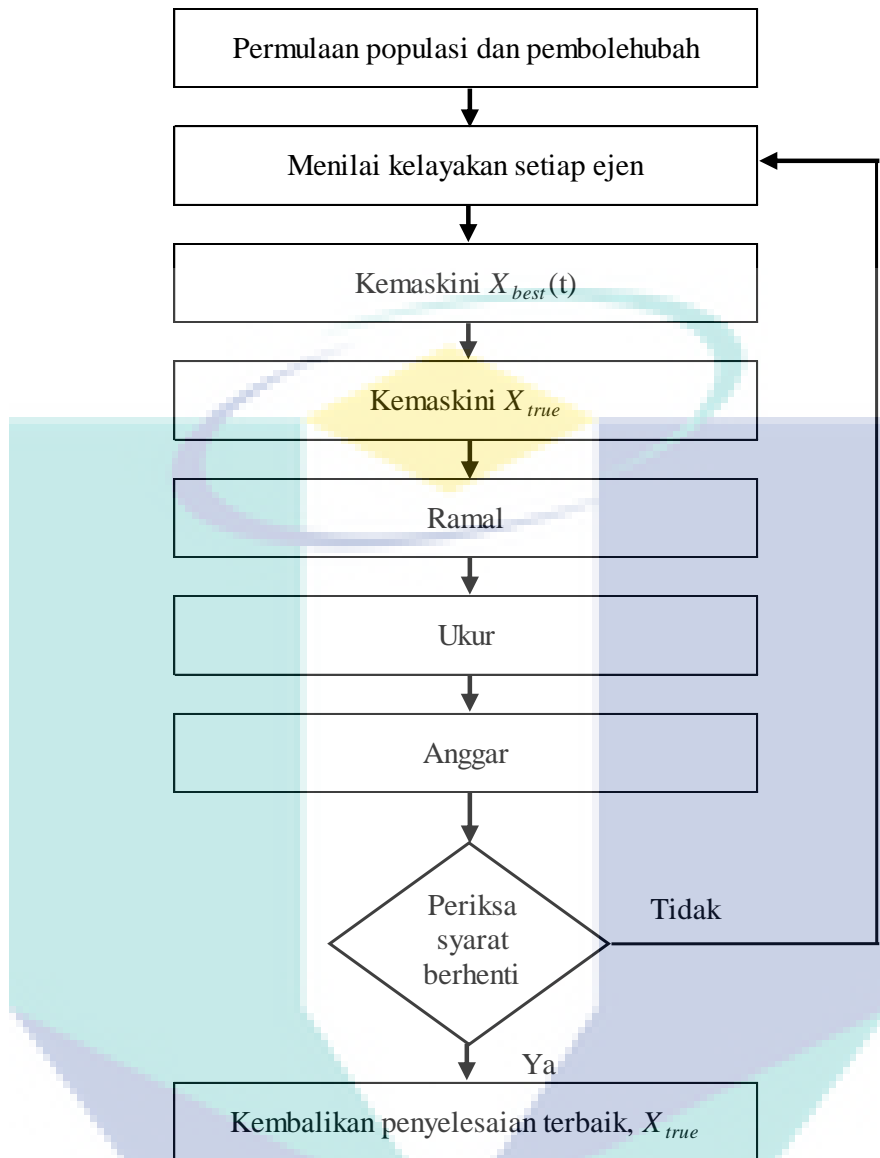
$$\mathbf{X}_i(t+1) = \mathbf{X}_i(t|t) + K(t) \times (\mathbf{Z}_i(t) - \mathbf{X}_i(t|t)) \quad (2.32)$$

$$P(t+1) = (1 - K(t)) \times P(t|t) \quad (2.33)$$

Akhir sekali, algoritma akan meneruskan proses carian sehingga syarat berhenti dicapai.

Oleh kerana SKF adalah algoritma pengoptimuman baru, maka sangat sedikit kertas kerja kajian yang melaporkan SKF untuk masalah pengoptimuman gabungan. BSKF, AMSKF dan DESKF dicadangkan oleh Md Yusof et al. pada tahun 2015 dan 2016.

UMP



Rajah 2.8 Carta aliran algoritma SKF

Sumber: Ibrahim, Z (2015)

## 2.11 Ringkasan

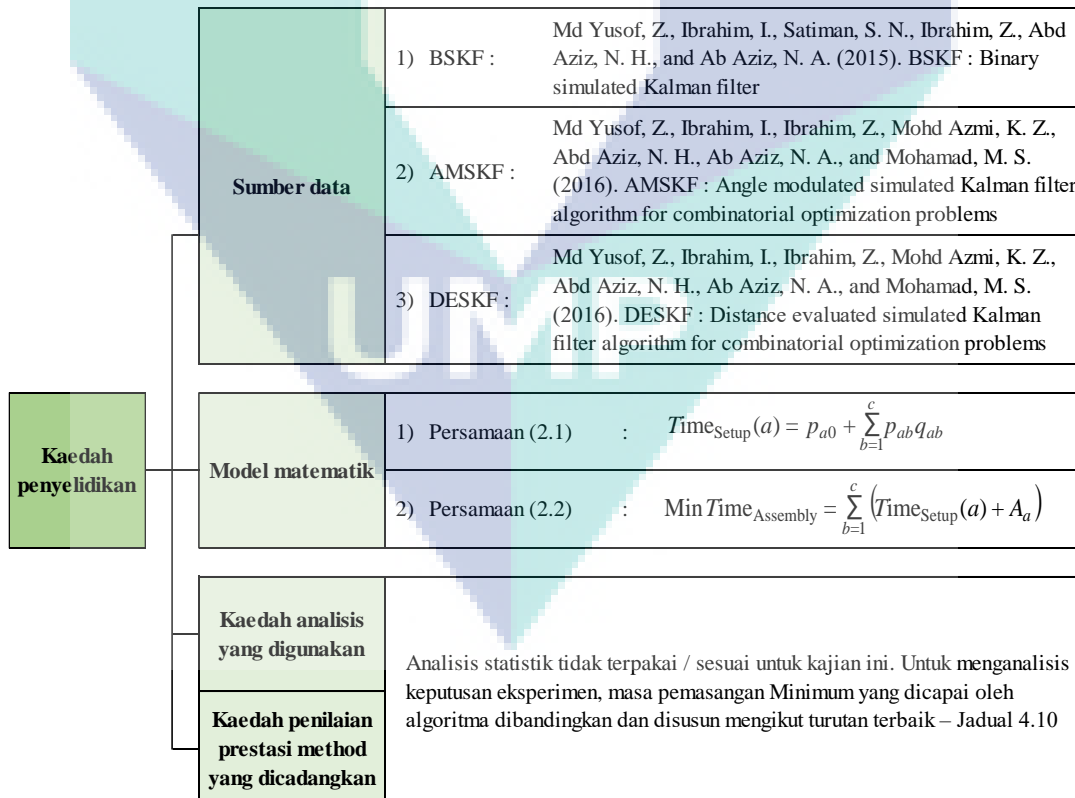
Bab ini membentangkan kajian terdahulu yang berkaitan untuk kajian ini. Ia merangkumi gambaran ringkas ASP. Seterusnya, kekangan ASP dan objektif pengoptimuman ASP diterangkan dengan jelas. Berkenaan dengan kajian literatur yang diberikan dalam bab ini, ia menunjukkan bahawa banyak kajian sebelumnya telah dicadangkan untuk menyelesaikan masalah pengoptimuman diskret. SKF adalah algoritma pengoptimuman baru yang mempunyai kelebihan meramal, mengukur dan menganggar dan ia belum pernah lagi diaplikasikan untuk ASP.

## BAB 3

### METODOLOGI

#### 3.1 Pengenalan

Bab ini akan dibentangkan menerangkan tentang pelaksanaan SKF lanjutan yang iaitu BSKF, AMSKF dan DESKF dan bagaimana algoritma-algoritma tersebut menyelesaikan masalah ASP. Seterusnya, menunjukkan parameter yang digunakan dalam algoritma BSKF, AMSKF dan DESKF untuk ASP. Carta alir kaedah penyelidikan yang dijalankan untuk kajian ini adalah seperti dalam Rajah 3.1.



Rajah 3.1 Carta alir kaedah penyelidikan yang dijalankan

### 3.2 Algoritma Penapis Kalman Simulasi Binari (BSKF)

Untuk menyelesaikan masalah pengoptimuman gabungan menggunakan SKF, istilah  $\Delta_i$  dalam Persamaan (3.1) dipetakan ke dalam nilai kebarangkalian [0,1]. Kemudian, nilai kebarangkalian dibandingkan dengan nombor rawak [0,1] untuk mengemaskini sedikit rentetan. Dalam BSKF, kebanyakan pengiraan adalah serupa dengan SKF asal. Pengubahsuaian hanya diperlukan semasa permulaan dan penjaanaan penyelesaian kepada masalah pengoptimuman gabungan.

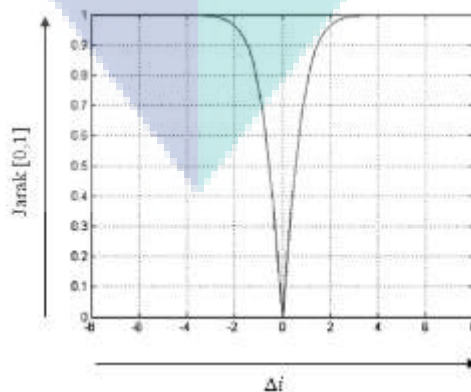
$$X_i(t + 1) = X_i(t|t) + \Delta_i \quad (3.1)$$

Semasa permulaan ejen, rentetan bit rawak,  $\Sigma_i$ , dijana untuk setiap ejen. Setiap bit dalam rentetan bit dikaitkan dengan dimensi. Panjang rentetan bit bergantung pada masalah dan tertakluk kepada panjang masalah. Dalam BSKF, fungsi yang ditunjukkan dalam Rajah 3.2 digunakan untuk nilai halaju ke dalam nilai kebarangkalian dalam jarak [0,1]. Istilah  $\Delta_i$  dipetakan ke nilai kebarangkalian dalam jarak [0,1] menggunakan fungsi pemetaan,  $S(\Delta_i(t))$ , seperti berikut dalam Persamaan (3.2)

$$S(\Delta_i(t)) = |\tanh \Delta_i(t)| \quad (3.2)$$

Selepas  $S(\Delta_i(t))$  dikira, nombor rawak,  $rand$ , dijana dan nilai binari pada dimensi  $d$  untuk ejen ke- $i$ ,  $\Sigma_i^d$ , dikemaskini mengikut peraturan dalam Persamaan (3.3)

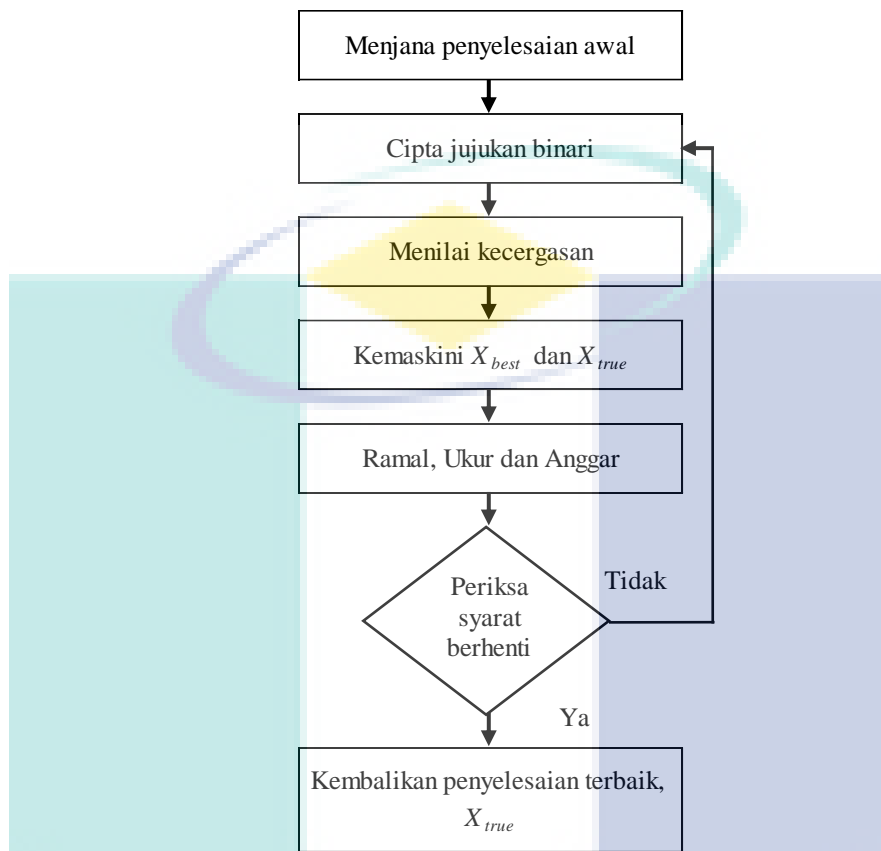
$$\begin{aligned} &\text{if } rand < S(\Delta_i(t)) \\ &\quad \text{then } \Sigma_i^d(t+1) = \text{complement } \Sigma_i^d(t) \\ &\quad \text{else } \Sigma_i^d(t+1) = \Sigma_i^d(t) \\ &\text{end} \end{aligned} \quad (3.3)$$



Rajah 3.2 Fungsi pemetaan



Rajah 3.3 menunjukkan carta alir bagi algoritma BSKF yang mengandungi langkah mencipta jujukan binari untuk mendapatkan penyelesaian masalah.



Rajah 3.3 Carta alir algoritma BSKF

### 3.3 Algoritma Penapis Kalman Diselakukan Modulasi Sudut (AMSKF)

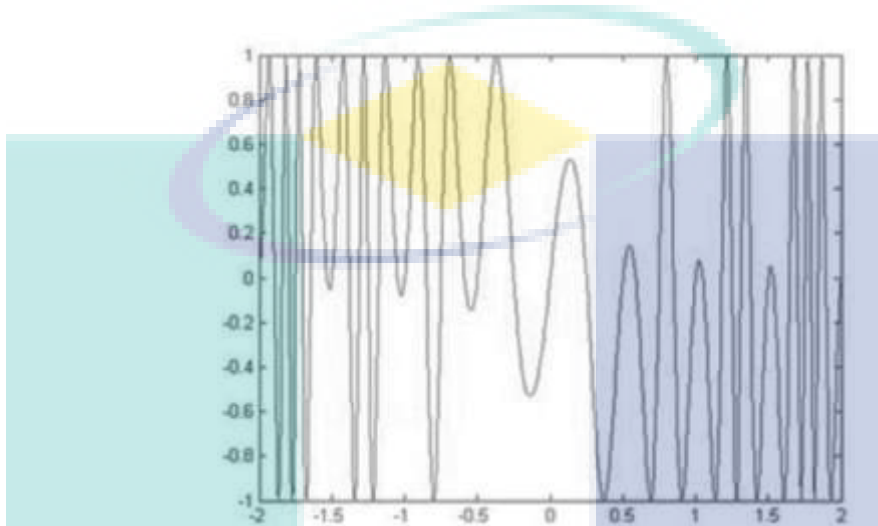
Idea utama kaedah sudut modulasi dalam menyelesaikan masalah pengoptimuman gabungan adalah dengan menggunakan fungsi,  $g(x)$ , untuk menghasilkan isyarat berterusan. Bentuk isyarat  $g(x)$  ditentukan oleh 4 pembolehubah, iaitu,  $a$ ,  $b$ ,  $c$ , dan  $d$ , seperti ditunjukkan dalam Persamaan (3.4).

$$g(x) = \sin(2\pi(x - a) x b \times \cos(A)) + d \quad (3.4)$$

di mana  $A = 2\pi(x - a) x c$

Rajah 3.4 menunjukkan contoh plot  $(x)$  untuk kes  $a = 0$ ,  $b = 1$ ,  $c = 1$ , dan  $d = 0$ . Rantau  $g(x) > 0$  dipanggil binari 1 rantau dan rantau  $g(x) < 0$  dipanggil rawak binari 0. Selepas itu, pensampelan berdasarkan masa pensampelan,  $T$ , dilaksanakan untuk menghasilkan rentetan bit dengan panjang  $n$ . Panjang yang diperlukan dalam rentetan bit

adalah bergantung kepada masalah dan ditentukan oleh saiz masalah pengoptimuman gabungan. Sebagai contoh, dalam masalah jurujual mengembara (*Travelling Salesman Problem* - TSP), jika saiz masalahnya adalah 10 bandar, maka 4 kali 10 bit diperlukan untuk mewakili penyelesaian TSP. Pada nilai  $x$  tertentu, jika  $g(x) > 0$ , bit 1 diberikan, sebaliknya, bit 0 diberikan.



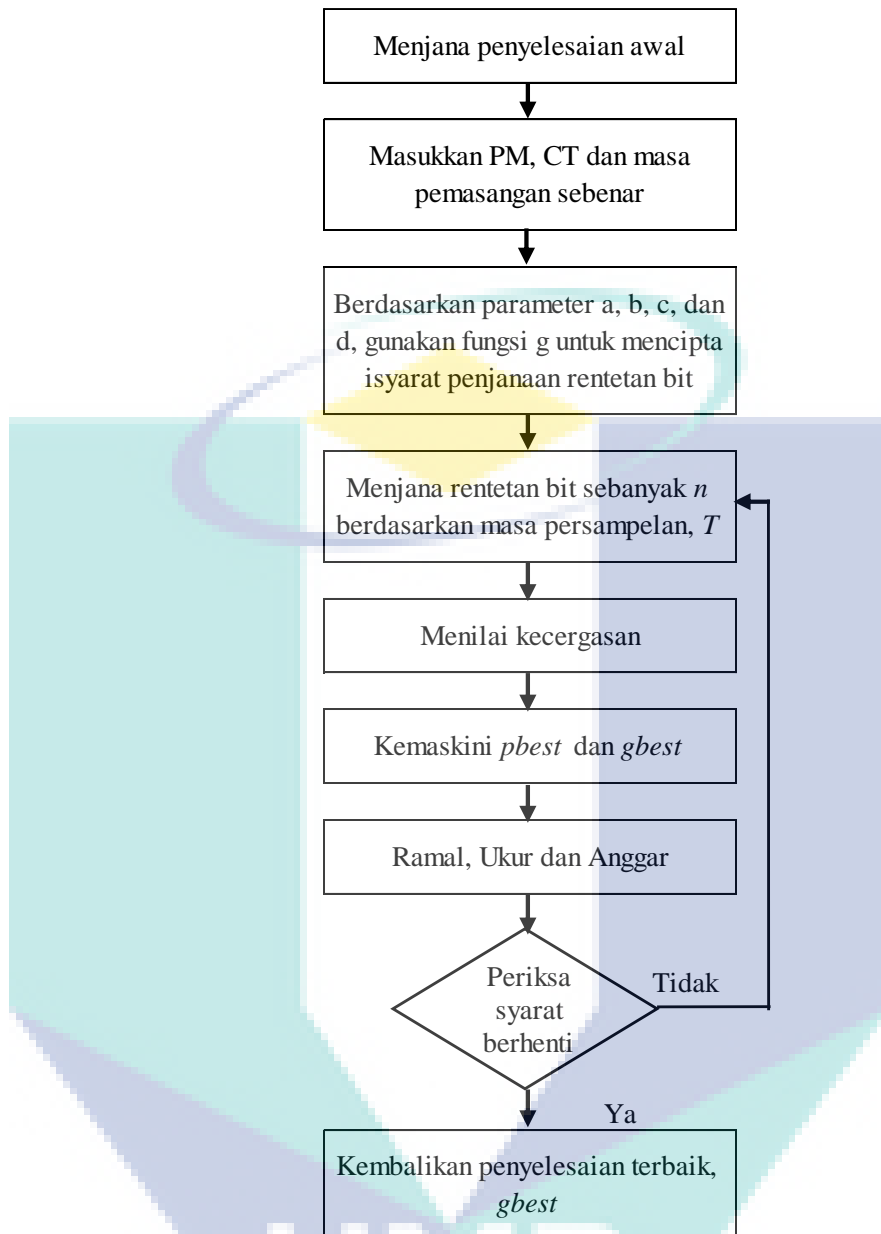
Rajah 3.4 Contoh plot  $g(x)$

Kelebihan utama pendekatan modulasi sudut ialah pengiraan yang kompleks dalam menghasilkan rentetan bit dimensi tinggi yang boleh dielakkan. Proses carian dalam menyelesaikan masalah pengoptimuman gabungan boleh dilakukan dengan mensimulasikan nilai-nilai  $a$ ,  $b$ ,  $c$ , dan  $d$  sahaja. Dalam kerja ini, penalaan dilakukan oleh algoritma SKF. Rajah 3.5 menunjukkan carta alir bagi algoritma AMSKF yang mengandungi langkah penggunaan fungsi  $g$  untuk mendapatkan penyelesaian masalah.

### 3.4 Algoritma Penapis Kalman Diselakukan Dinilai Jarak (DESKF)

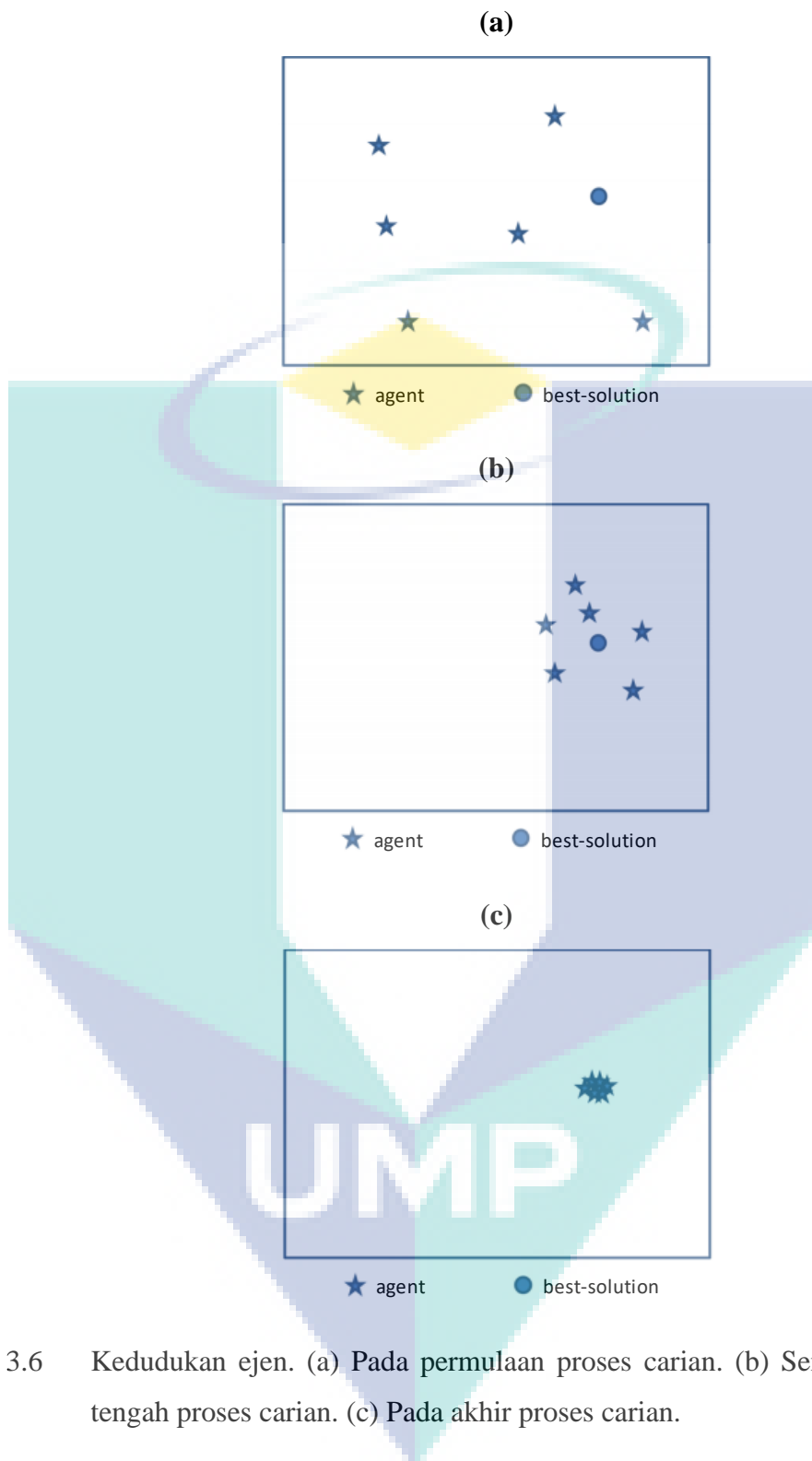
Dalam algoritma carian berasaskan populasi, secara amnya, ejen secara rawak diletakkan di ruang carian. Kemudian, ejen bergerak di ruang carian untuk mencari minimum atau maksimum global. Kedudukan ejen dalam ruang carian semasa proses carian tipikal digambarkan dalam Rajah 3.6

- (a) Semasa permulaan carian, penerokaan lebih cenderung berlaku untuk memastikan carian meliputi hampir semua kawasan di ruang carian. Dalam proses pencarian tahap ini, kedudukan antara ejen biasanya jauh antara satu sama lain.



Rajah 3.5 Carta alir algoritma AMSKF

- (b) Memandangkan proses carian berterusan, pada akhir pencarian, penerokaan tidak lagi dipilih kerana penalaan atau eksploitasi adalah lebih baik pada masa tersebut.
- (c) Semasa eksploitasi, ejen semakin dekat antara satu sama lain dan dengan itu, jarak di antara mereka berkurangan.



Rajah 3.6 Kedudukan ejen. (a) Pada permulaan proses carian. (b) Semasa tengah-tengah proses carian. (c) Pada akhir proses carian.

Biasanya, apabila lelaran berterusan, jarak antara agen dan penyelesaian terbaik yang begitu jauh telah berkurangan. Jarak ini memainkan peranan penting dalam algoritma penapis Kalman diselakukan dinilai jarak yang dicadangkan (DESKF). Dalam DESKF, jarak dipetakan ke dalam nilai kebarangkalian  $[0,1]$  dan kemudian nilai

kebarangkalian akan dibandingkan dengan nombor rawak [0,1] untuk mengemaskini sedikit rentetan atau penyelesaian kepada masalah pengoptimuman gabungan.

Secara terperinci, kebanyakan pengiraan dalam DESKF yang dicadangkan adalah serupa dengan SKF asal. Pengubahsuaian hanya diperlukan semasa permulaan dan penjaanaan penyelesaian kepada masalah pengoptimuman gabungan.

Semasa permulaan ejen, dalam SKF, keadaan setiap ejen diberi secara rawak. Permulaan tambahan diperkenalkan di DESKF. Setiap ejen dikaitkan dengan rentetan bit rawak juga. Panjang rentetan bit bergantung kepada masalah dan tertakluk kepada saiz masalah. Oleh itu, 2 jenis pembolehubah dikaitkan dengan ejen dalam SKF. Pembolehubah berterusan,  $\mathbf{X}$ , yang dihasilkan sebagai nilai anggaran SKF (juga sama dengan kedudukan ejen dalam ruang pencarian), dan sedikit string,  $\Sigma$ , yang digunakan untuk mewakili penyelesaian kepada masalah pengoptimuman gabungan.

Di DESKF, untuk dimensi yang ke- $D$  tertentu, jarak antara ejen yang ke- $i$  kepada larutan yang paling baik pada lelaran  $T$  boleh dikira seperti dalam Persamaan (3.5)

$$D_i^d(t) = x_i^d(t) - x_{best-so-far}^d(t) \quad (3.5)$$

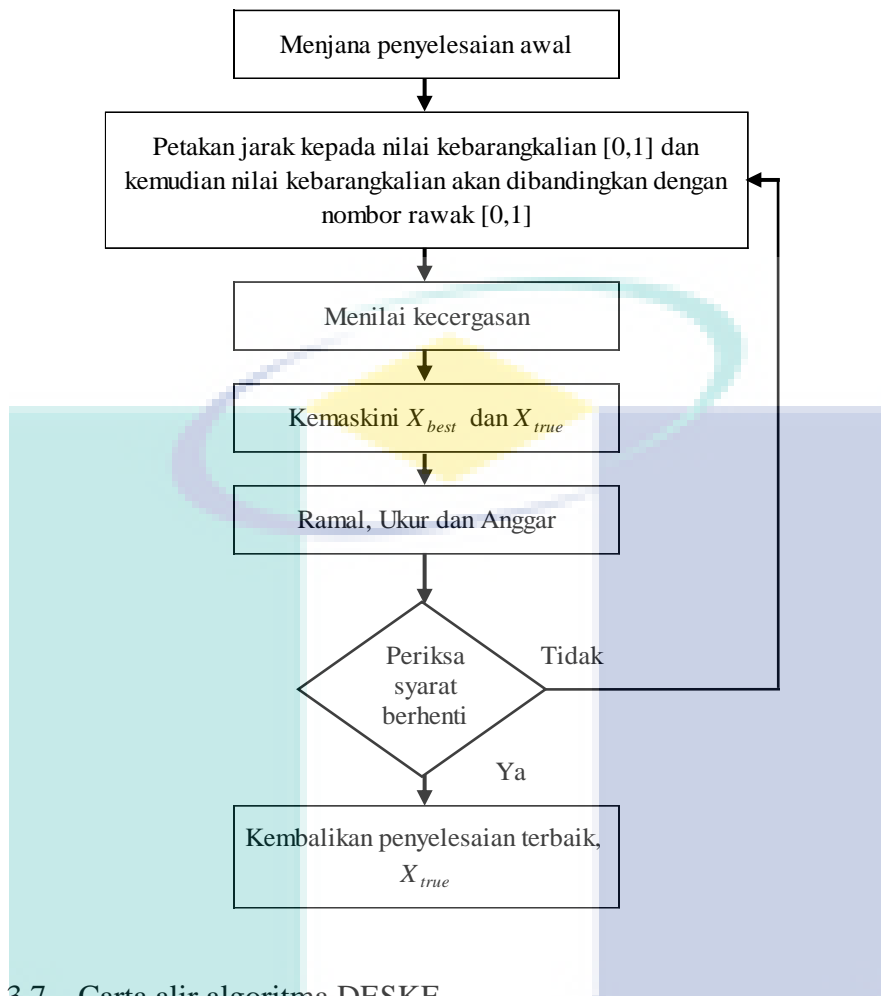
Dalam DESKF, satu fungsi seperti yang ditunjukkan dalam Rajah 3.1, digunakan untuk memetakan nilai halaju ke dalam nilai kebarangkalian dalam jarak [0,1]. Nilai jarak,  $D_i^d(t)$ , dipetakan kepada nilai kebarangkalian dalam jarak [0,1] menggunakan fungsi kebarangkalian,  $S(D_i^d(t))$ , seperti dalam Persamaan (3.6)

$$S(D_i^d(t)) = |\tanh(D_i^d(t))| \quad (3.6)$$

Selepas  $S(D_i^d(t))$  dikira, nombor rawak,  $rand$ , dihasilkan dan nilai binari pada dimensi  $d$  ejen ke- $i$ ,  $\Sigma$ , dikemaskini mengikut peraturan seperti dalam Persamaan (3.7)

$$\begin{aligned} &\text{if } rand < S(D_i^d(t)) \\ &\quad \text{then } \Sigma_i^d(t+1) = \text{complement } \Sigma_i^d(t) \\ &\quad \text{else } \Sigma_i^d(t+1) = \Sigma_i^d(t) \\ &\text{end} \end{aligned} \quad (3.7)$$

Rajah 3.7 menunjukkan carta alir bagi algoritma DESKF yang mengandungi langkah memetakan jarak kepada nilai kebarangkalian [0,1] untuk penyelesaian masalah.



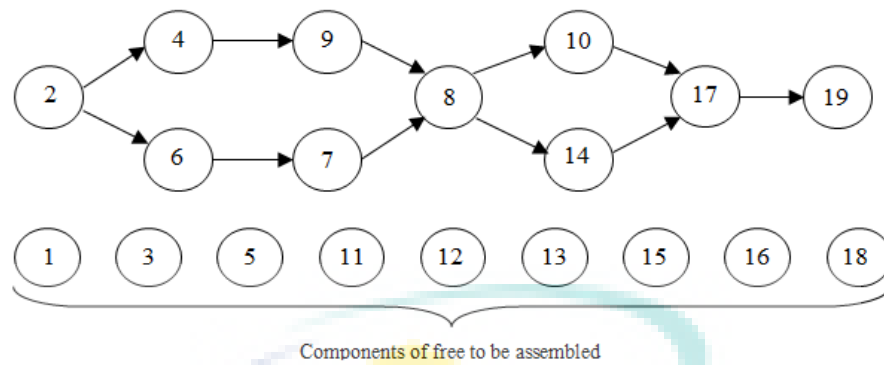
Rajah 3.7 Carta alir algoritma DESKF

### 3.5 Aplikasi Algoritma SKF Lanjutan dalam ASP

Dalam kajian ini, masalah perancangan jujukan pemasangan diselesaikan menggunakan SKF. Objektifnya adalah untuk menjana jujukan pemasangan optimum untuk meminimumkan masa pengeluaran. Pada masa yang sama, kos pengeluaran juga boleh dijimatkan. Bahagian ini membentangkan pendekatan yang dicadangkan berdasarkan BSKF, AMSKF, dan DESKF yang diaplikasikan dalam ASP. Perwakilan vektor yang sepadan dengan komponen pemasangan zarah diwakili dalam Rajah 3.8; dalam kes ini, jujukan penyelesaian adalah 2-3-1-5-4. Panjang rentetan bergantung pada jumlah komponen yang digunakan dalam proses pemasangan.

2	3	1	5	4
---	---	---	---	---

Rajah 3.8 Contoh jujukan pemasangan yang diwakili oleh zarah



Rajah 3.9 PD untuk kajian kes

Sumber: Choi, Lee, & Cho (2008)

Rajah 3.9 menunjukkan PD bagi pemasangan produk berdasarkan hipotesis dengan 19 komponen bagi kajian kes (Choi, Lee, & Cho, 2008). Dalam Rajah ini, komponen yang bebas untuk dipasang ialah komponen yang boleh dipasangkan tanpa mengira mana-mana bahagian jujukan. Rajah keutamaan boleh diterjemahkan ke dalam PM seperti yang ditunjukkan dalam Jadual 3.1, dimana Komp. (b) dipasang terlebih dahulu sebelum Komp. (a).

Kadang kala, beberapa komponen pemasangan tidak dapat disepadukan ke jujukan pemasangan yang layak. Penentuan komponen pemasangan yang tidak sepadan dengan jujukan pemasangan yang layak dicapai dengan memuaskan semua kekangan PM antara komponen dalam perhimpunan yang ditentukan sebelumnya, sama ada dari CAD atau analisis pembongkaran (Gao, Qian, Li, & Wang, 2009).

Jadual pekali yang diwakili dalam matriks kemudiannya digunakan untuk mengira kecergasan setiap zarah yang diwakili oleh jujukan pemasangan yang layak. CT (S. Motavalli, 1997) yang telah digunakan sebagai rujukan dalam kajian ASP terdahulu telah ditunjukkan dalam Jadual 3.2, dimana Komp. (b) dipasang terlebih dahulu sebelum Komp. (a). Berhubung dengan kecergasan setiap zarah yang dikira, kedudukan peribadi setiap zarah, *pbest* bagi setiap zarah akan ditetapkan pada kedudukannya. Nilai *pbest* terbaik yang dinamakan sebagai terbaik global, *gbest* kemudian ditetapkan. Dalam kajian ini, masa pemasangan ditetapkan dalam unit masa seperti dalam kajian yang terdahulu, iaitu 1 unit masa bersamaan dengan 1024 mikro saat.

Jadual 3.1 PM untuk kajian kes.

Komp. (a)	Komp. (b)																		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1																			
2																			
3																			
4		1																	
5																			
6		1																	
7		1				1													
8		1		1		1	1			1									
9		1		1															
10		1		1		1	1			1	1								
11																			
12																			
13																			
14		1		1		1	1			1	1								
15																			
16																			
17		1		1		1	1			1	1								1
18																			
19		1		1		1	1			1	1								1

Sumber: Choi, Lee, & Cho (2008)



Jadual 3.2 CT bagi pelbagai komponen dalam pemasangan.

Komp. (a)	Komp. (b)																		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	10	1	2	3	4	5	6	7	8	9	3.2	4.3	7	6.1	1.2	3.4	0	0	7.4
2	1.5	10	2	2	2	2	2	2	2	2	0	3.1	6	4.3	2.7	4.8	0	3	0.5
3	1	2.3	10	0	4	5	0	4	2.3	4.3	9.8	2.4	5	1.2	3.4	4.5	5.6	3.4	3.1
4	0	2	3.4	10	4.5	0	4	0	8	0	3.4	5.6	5	0	0	3.4	0	0	9.8
5	1.2	1	2	3	10	7.9	8.9	0	1.2	2	2.3	0	3	0	3.6	0	2.8	9.8	0
6	9.8	4.5	0	1.2	3.6	10	3.4	4	0	2.3	4.6	5.6	0	4	3	2	0	0.4	3.2
7	0.5	1.4	2.3	0.5	1.9	1	10	13.4	1.2	4	2.3	0	3	5.7	8.3	2	0.1	0	0.5
8	0	0	0	0	0	1.8	9.8	10	2.3	3	8.9	2.3	0	0	2.3	0.5	9.8	0	2.3
9	1	3	4.5	2.3	4.6	9.8	7.5	6.8	10	6	2.3	3.4	5	12.3	3.4	5.61	1	0	0
10	2.3	4.5	2.3	0	2.3	0	2.1	0	4.5	10	1.1	2.2	2	0	0	2.1	1.2	5.4	9.2
11	1	1	2	3	4	5	6	7	8	9	10	4.5	3	6.1	1.2	3.4	0.3	0	1.3
12	1.5	0	2	2	2	2	2	1	2	2	11.2	10	6	4.3	2.7	4.8	0	3	0.5
13	1	2.3	0	0	4	5	0	4	2.3	4.3	9.8	2.4	10	1.2	2.4	4.5	1.6	2.4	3.1
14	0	2	3.4	0	4.5	0	4	0	8	0	3.4	5.6	5	10	2.1	1.4	1	0	2.8
15	1.2	1	2	3	0	7.9	8.9	0	1.2	2	1.3	4	3	1.4	10	1.3	9.8	9.8	2
16	9.8	4.5	0	1.2	3.6	0	3.4	4	0	2.3	4.6	3.6	0	4	3	10	1.5	0	3.2
17	1	3	4	5	0	5	4	3.4	1.2	4	1.3	0	2	3.7	4.3	2.3	10	3.8	10
18	0.6	0.5	3.4	1.2	3	2	9.8	2	2.3	3	5.9	2.3	0	1	2.3	0.5	9.8	10	2.3
19	1	3	4.5	2.3	4.6	9.8	7.5	6.8	0	6	3.3	3	2	3.3	4.4	2.6	0.3	2.5	10

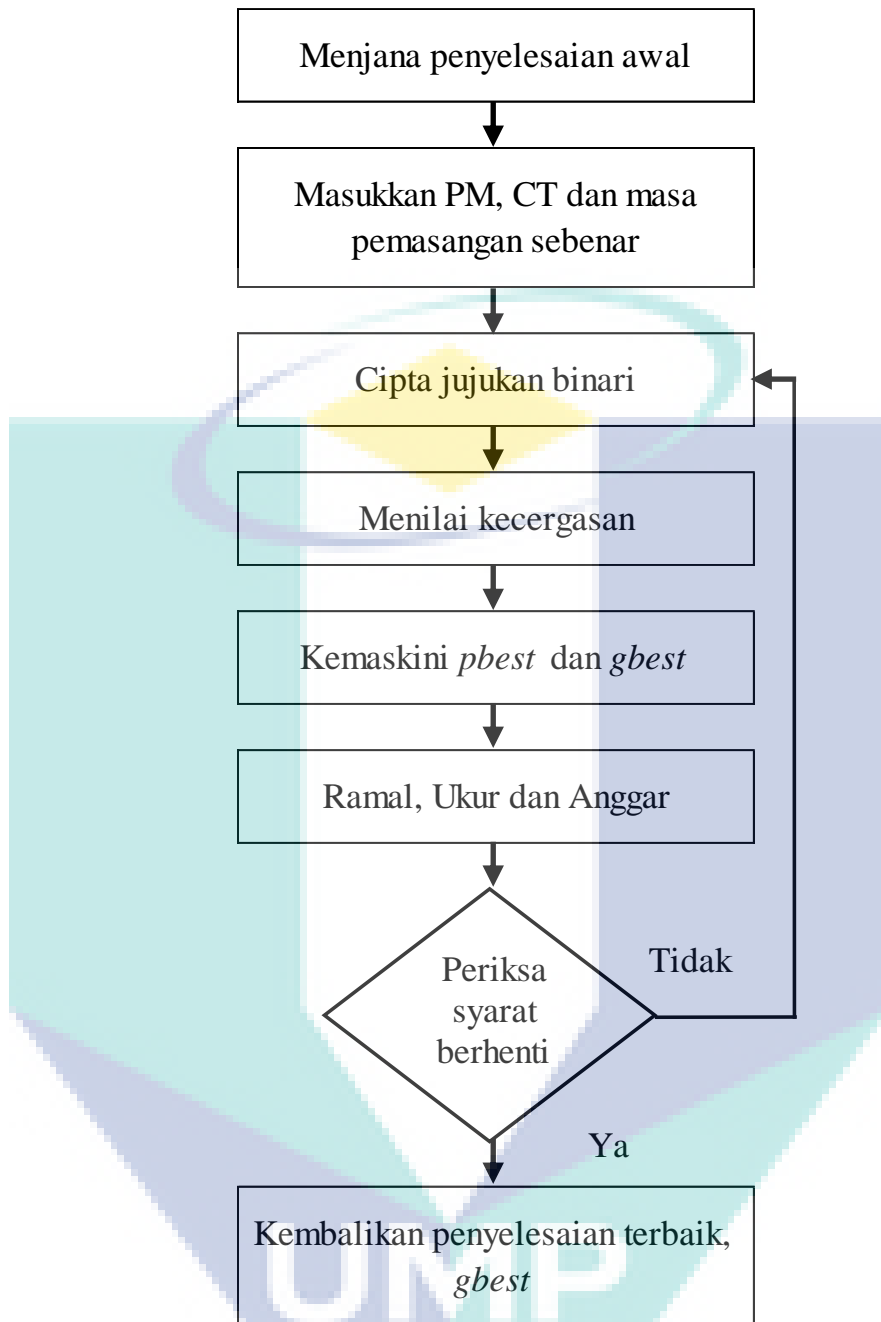
Sumber: Choi, Lee, & Cho (2008)

Untuk mengesahkan penghasilan satu jujukan pemasangan yang boleh dilaksanakan bagi setiap zarah, jujukan pemasangan yang dikemas kini bagi setiap zarah yang dihasilkan oleh proses kedudukan pengemaskinian akan berubah menjadi jujukan pemasangan layak. Untuk memasang komponen produk dengan cara yang sah, hanya jujukan pemasangan yang layak yang harus digunakan. Kemungkinan jujukan dapat ditentukan dengan merujuk kepada PM. Berdasarkan PM, algoritma yang dapat mengembangkan jujukan pemasangan yang diperbaiki dari setiap zarah ke jujukan pemasangan yang layak akan dikembangkan. Rajah 3.10, Rajah 3.11, dan Rajah 3.12 menunjukkan proses pembentukan jujukan pemasangan setiap zarah atau ejen yang boleh dilaksanakan untuk BSKF, AMSKF, dan DESKF masing-masing.

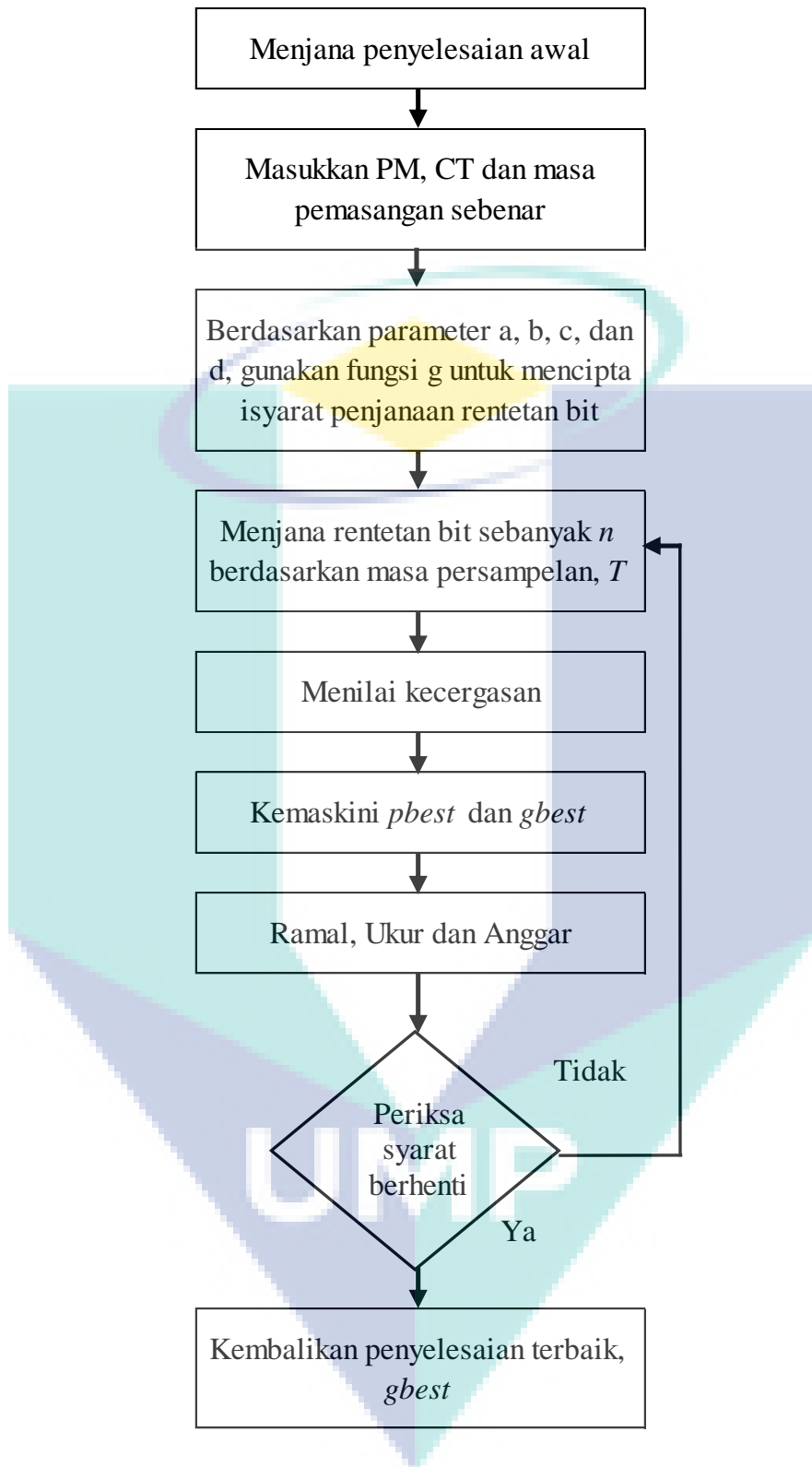
Dalam algoritma ini, kedudukan setiap jujukan ditukar secara rawak antara komponen yang tidak mungkin dan komponen yang boleh dilaksanakan. Proses penukaran berakhir apabila setiap komponen berlaku dalam jujukan pemasangan di mana jujukan itu dapat dilaksanakan.

Penilaian kecergasan bagi setiap zarah dilaksanakan untuk meningkatkan nilai objektif. Oleh itu, bagi setiap lalaran, setiap zarah yang diwakili oleh jujukan pemasangan layak dilaksanakan.  $pbest$  dari setiap zarah kemudian dikemas kini oleh nilai objektif yang lebih tinggi, yang diperoleh dari posisi baru zarah  $pbest$ .  $gbest$  kemudian dikemas kini oleh  $pbest$  dengan nilai objektif tertinggi. Penyelesaian optimum ASP yang diwakili oleh  $gbest$  dihasilkan apabila syarat berhenti akhirnya dipenuhi. Selepas keadaan berhenti dipenuhi, prestasi kaedah yang dicadangkan berdasarkan algoritma boleh disiasat.

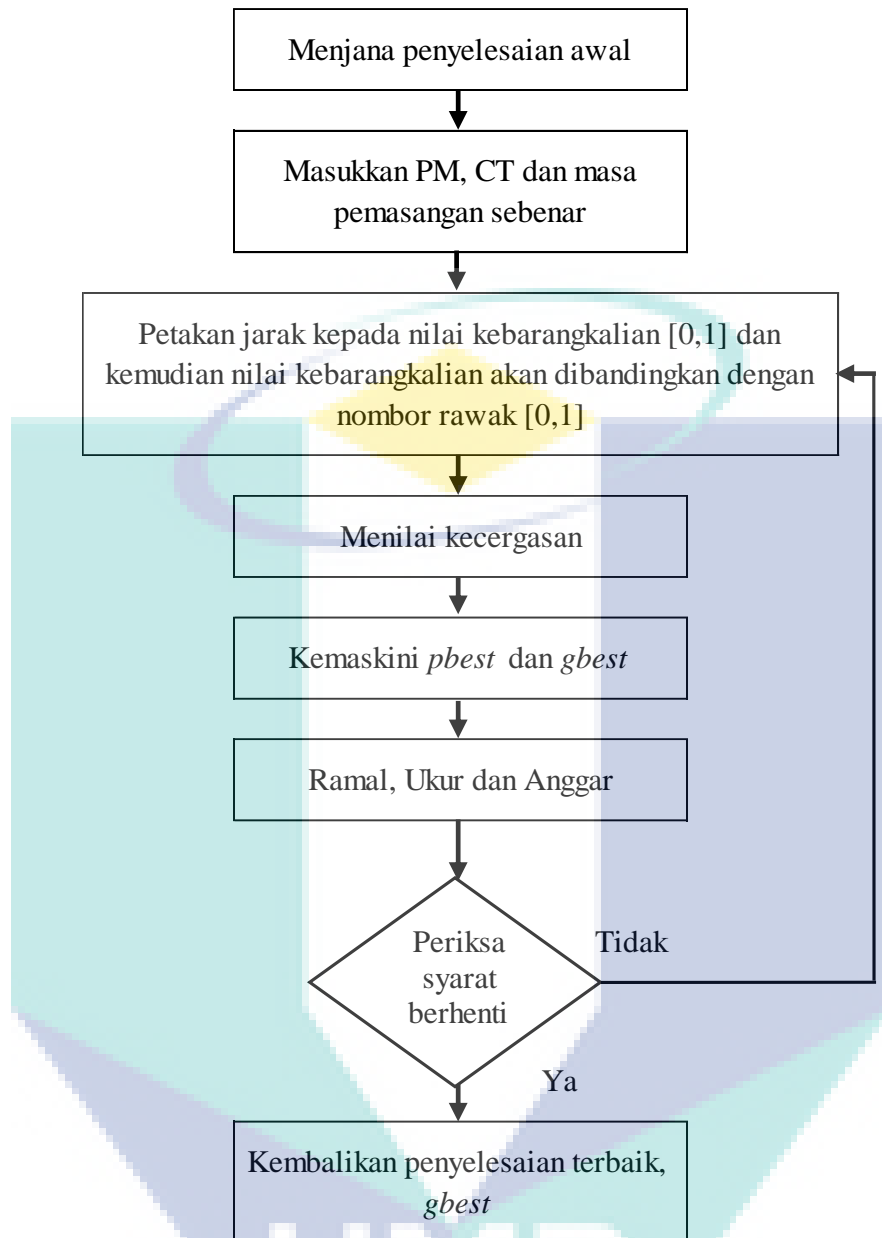
Untuk menilai kecergasan bagi setiap ejen, jumlah masa pemasangan harus diperolehi. Waktu pemasangan sepenuhnya adalah gabungan dari masa persediaan dan masa pemasangan sebenar. Adalah diandaikan bahawa tanpa mengira jujukan pemasangan, masa perhimpunan sebenarnya adalah malar, dan alat dan persediaan yang sesuai untuk setiap komponen untuk dipasang diperlukan. Kedua-dua item bergantung kepada geometri komponen itu sendiri dan komponen dipasang sehingga selesai.



Rajah 3.10 Proses pembentukan jujukan pemasangan setiap zarah untuk BSKF dalam ASP



Rajah 3.11 Proses pembentukan jujukan pemasangan setiap zarah untuk AMSKF



Rajah 3.12 Proses pembentukan jujukan pemasangan setiap zarah untuk DESKF

### 3.6 Parameter bagi BSKF, AMSKF, dan DESKF untuk ASP

Dalam eksperimen, parameter yang digunakan untuk kaedah yang dicadangkan berdasarkan BSKF, AMSKF, dan DESKF dengan 10, 20, 30 ejen, untuk 1,000 dan 5,000 lelaran dibentangkan dalam Jadual 3.3. Nilai anggaran kovarians ralat awal, bunyi bising dan bunyi pengukuran dipilih berdasarkan nilai piawaian yang digunakan untuk SKF.

Jadual 3.3 Parameter eksperimen untuk pendekatan yang dicadangkan berdasarkan BSKF, AMSKF, dan DESKF dengan 10, 20, 30 agen, untuk 1,000 dan 5,000 lelaran.

Parameter	BSKF	AMSKF	DESKF
Bilangan pelaksanaan	50	50	50
Anggaran kovarians ralat awal, $P(0)$	100	100	100
Hingar proses, $Q$	0.5	0.5	0.5
Hingar pengukuran, $R$	0.5	0.5	0.5

### 3.7 Ringkasan

Bab ini memberikan penjelasan tentang algoritma Penapis Kalman dan Penapis Kalman Diselakukan. Kemudian, SKF lanjutan iaitu BSKF, AMSKF dan DESKF yang dicadangkan untuk digunakan pada ASP dibentangkan. Berdasarkan jurang kajian yang telah dilaporkan dalam kajian literatur, kajian ini dilaksanakan dengan tujuan untuk mengaplikasikan SKF ke ASP dan menganalisis pencapaiannya untuk menyelesaikan masalah ASP untuk mendapatkan penyelesaian optimum untuk meminimumkan masa pemasangan.

## BAB 4

### KEPUTUSAN DAN PERBINCANGAN

#### 4.1 Pengenalan

Dalam bab ini, prestasi beberapa eksperimen yang dijalankan ke atas algoritma BSKF, AMSKF, dan DESKF pada ASP akan dibentangkan. Dalam menyiasat prestasi algoritma untuk ASP, 19 komponen telah dipilih sebagai kajian kes (Motavalli & Islam, 1997, Choi et al., 2008 dan Mukred et al., 2012). Setiap penyelesaian adalah dalam bentuk jujukan pemasangan yang layak. Jujukan optimum kemudian dipilih dari jujukan pemasangan yang layak dengan menilai kecergasan setiap penyelesaian. Yang terbaik dari populasi adalah jujukan yang lebih optimum sehingga syarat keadaan berhenti dipenuhi. Selepas keadaan berhenti dipenuhi, prestasi kaedah yang dicadangkan boleh disiasat. Untuk membuat keputusan, prestasi kaedah yang dicadangkan menggunakan BSKF, AMSKF, dan DESKF dibandingkan dengan 6 kaedah yang berkaitan dengan metaheuristik. Nilai min biasanya difokuskan dalam analisis keputusan untuk mengenal pasti masa pemasangan minimum yang boleh diperolehi dengan menggunakan parameter percubaan.

Bab ini memberikan kecekapan relatif prestasi kaedah yang dicadangkan berdasarkan BSKF, AMSKF, dan DESKF. Setiap algoritma yang dicadangkan kemudiannya dibandingkan dengan kaedah berdasarkan BGSA (Rashedi et al., 2009), BPSO (Mukred et al., 2012), MSPSO (Ibrahim et al., 2016), MSPSOER (Ibrahim et al., 2016), MSGSA (Ibrahim et al., 2016), dan MSGSAER (Ibrahim et al., 2016). Dalam semua perbandingan, kualiti keputusan setiap pendekatan yang dicadangkan diukur berdasarkan nilai kecergasan penyelesaian terbaik dalam meminimumkan jumlah masa pemasangan.

## 4.2 Keputusan Aplikasi BSKF, AMSKF, dan DESKF untuk ASP

Kualiti hasil kaedah yang dicadangkan berdasarkan BSKF, AMSKF, dan DESKF untuk 1,000 lelaran, masing-masing dibentangkan dalam Jadual 4.1, Jadual 4.2 dan Jadual 4.3. Sementara itu, kualiti untuk kaedah yang dicadangkan berdasarkan BSKF, AMSKF, dan DESKF untuk 5,000 lelaran, masing-masing dibentangkan dalam Jadual 4.4, Jadual 4.5 dan Jadual 4.6. Keputusan eksperimen ini adalah merangkumi penyelesaian terbaik (Min), penyelesaian purata (Mean), penyelesaian terburuk (Max), dan sisihan piawai (SD) dalam menyelesaikan masalah ASP. Sisihan piawai adalah ukuran yang digunakan untuk mengukur jumlah variasi set nilai data. Sisihan piawai yang rendah menunjukkan bahawa titik data cenderung dekat dengan nilai jangkaan set, yang bermaksud masa pemasangan adalah minimum. Sementara itu, sisihan piawai yang tinggi menunjukkan bahawa titik data tersebar dalam pelbagai nilai yang lebih luas.

Jadual 4.1 Keputusan untuk kaedah yang dicadangkan berdasarkan kepada BSKF untuk 1,000 lelaran.

<b>Bilangan ejen</b>	<b>Minimum</b>	<b>Purata</b>	<b>Maksimum</b>	<b>SD</b>
10	523.70	542.38	564.00	9.87
20	524.90	538.54	551.30	6.35
30	512.70	535.63	549.21	7.66

Jadual 4.2 Keputusan untuk kaedah yang dicadangkan berdasarkan kepada AMSKF untuk 1,000 lelaran.

<b>Bilangan ejen</b>	<b>Minimum</b>	<b>Purata</b>	<b>Maksimum</b>	<b>SD</b>
10	504.70	524.63	534.70	5.57
20	503.40	521.68	529.70	5.53
30	503.91	520.19	528.70	4.94

Jadual 4.3 Keputusan untuk kaedah yang dicadangkan berdasarkan DESKF untuk 1,000 lelaran.

<b>Bilangan ejen</b>	<b>Minimum</b>	<b>Purata</b>	<b>Maksimum</b>	<b>SD</b>
10	515.20	522.68	530.70	3.57
20	514.40	521.74	526.90	3.09
30	512.40	520.23	527.00	3.65



Jadual 4.4 Keputusan untuk kaedah yang dicadangkan berdasarkan kepada BSKF untuk 5,000 lelaran.

<b>Bilangan ejen</b>	<b>Minimum</b>	<b>Purata</b>	<b>Maksimum</b>	<b>SD</b>
10	523.90	542.40	560.71	8.06
20	518.11	535.13	552.01	7.56
30	519.90	533.45	544.60	5.90

Jadual 4.5 Keputusan untuk kaedah yang dicadangkan berdasarkan AMSKF untuk 5,000 lelaran.

<b>Bilangan ejen</b>	<b>Minimum</b>	<b>Purata</b>	<b>Maksimum</b>	<b>SD</b>
10	503.80	517.31	526.40	5.12
20	504.80	514.39	524.20	4.98
30	506.20	513.49	522.20	3.99

Jadual 4.6 Keputusan untuk kaedah yang dicadangkan berdasarkan kepada DESKF untuk 5,000 lelaran.

<b>Bilangan ejen</b>	<b>Minimum</b>	<b>Purata</b>	<b>Maksimum</b>	<b>SD</b>
10	503.80	518.91	524.80	4.30
20	511.20	517.57	522.80	3.16
30	508.60	515.42	519.20	2.51

Jadual 4.7 menunjukkan kompilasi keputusan terbaik kaedah yang dicadangkan berdasarkan BSKF, AMSKF, dan DESKF dengan dengan bilangan ejen 10, 20 dan 30 serta bilangan lelaran 1000 dan 5000.

Jadual 4.7 Kompilasi keputusan terbaik kaedah yang dicadangkan berdasarkan BSKF, AMSKF, dan DESKF

<b>Bilangan ejen</b>	<b>BSKF</b>		<b>AMSKF</b>		<b>DESKF</b>	
	<b>1000</b>	<b>5000</b>	<b>1000</b>	<b>5000</b>	<b>1000</b>	<b>5000</b>
10	523.70	523.90	504.70	503.80	515.20	503.80
20	524.90	518.11	503.40	504.80	514.40	511.20
30	512.70	519.90	503.91	506.20	512.40	508.60

Jadual 4.8 menunjukkan perbandingan keputusan terbaik kaedah yang dicadangkan berdasarkan BSKF, AMSKF, dan DESKF dengan jujukan pemasangan mereka. Ini menunjukkan bahawa kaedah yang dicadangkan berdasarkan AMSKF mengatasi kaedah berdasarkan BSKF, dan DESKF dengan menghasilkan penyelesaian optimum yang minimum untuk Min dan Mean.

Jadual 4.8 Keputusan terbaik dan jujukan pemasangan yang berkaitan dengan kaedah yang dicadangkan berdasarkan BSKF, AMSKF, dan DESKF.

Kaedah	Bilangan lelaran	Bilangan ejen	Jumlah masa pemasangan		Jujukan pemasangan
			Minimum	Purata	
AMSKF	1,000	20	503.40	521.68	1-2-12-4-3-9-13-15-11-5-16-6-18-7-8-14-10-17-19
AMSKF	5,000	10	503.80	517.31	1-4-1-12-3-9-13-15-11-5-16-6-18-7-8-14-10-17-19
DESKF	5,000	10	503.80	518.91	2-1-12-4-3-9-13-5-15-11-16-6-18-7-8-14-10-17-19
AMSKF	1,000	30	503.91	520.19	1-3-2-4-9-12-13-15-11-5-16-6-18-7-8-14-10-17-19
AMSKF	1,000	10	504.70	524.63	2-1-12-4-3-9-13-15-11-5-16-6-18-7-8-14-10-17-19
AMSKF	5,000	20	504.80	514.39	1-2-3-4-12-9-13-15-11-5-16-6-18-7-8-14-10-17-19
AMSKF	5,000	30	506.20	513.49	1-2-12-4-3-9-13-15-11-5-16-6-7-8-18-14-10-17-19
DESKF	5,000	30	508.60	515.42	1-2-12-9-3-4-13-15-11-5-16-6-18-7-8-14-10-17-19
DESKF	5,000	20	511.20	517.57	1-2-3-4-12-9-13-15-11-5-16-6-18-7-8-10-14-17-19
DESKF	1,000	30	512.40	520.23	1-12-2-3-4-9-13-15-11-5-16-6-18-7-8-14-10-17-19
BSKF	1,000	30	512.70	535.63	2-1-12-4-3-9-13-15-11-5-6-16-18-7-8-14-10-17-19
DESKF	1,000	20	514.40	521.74	1-2-4-3-12-9-13-15-11-5-16-6-18-7-8-10-14-17-19

Jadual 4.8 Sambungan.

Kaedah	Bilangan lelaran	Bilangan ejen	Jumlah masa pemasangan		Jujukan pemasangan
			Minimum	Purata	
DESKF	1,000	10	515.20	522.68	1-2-12-4-3-9-13-15-11-5-16-6-7-8-18-14-10-17-19
BSKF	5,000	20	518.11	535.13	2-1-12-4-3-9-13-15-11-5-16-6-18-7-8-14-10-17-19
BSKF	5,000	30	519.90	533.45	1-4-1-12-3-9-13-15-11-5-16-6-18-7-8-14-10-17-19
BSKF	1,000	10	523.70	542.38	1-3-2-4-9-12-13-15-11-5-16-6-18-7-8-14-10-17-19
BSKF	5,000	10	523.90	542.40	2-1-12-4-3-9-13-15-11-5-16-6-18-7-8-14-10-17-19
BSKF	1,000	20	524.90	538.54	2-1-12-4-3-9-13-5-11-15-16-6-18-7-8-14-10-17-19

#### 4.3 Keputusan Aplikasi BSKF, AMSKF, dan DESKF dibandingkan dengan algoritma lain untuk ASP

Jadual 4.9 menunjukkan parameter yang digunakan untuk kaedah yang dicadangkan berdasarkan BPSO, MSPSO, MSPSOER, BGSA, MSGSA dan MSGSAER. Parameter-parameter ini dipilih selepas satu siri eksperimen dijalankan untuk menyesuaikan parameter terbaik untuk algoritma yang dicadangkan bagi masalah perancangan jujukan pemasangan kerana kejayaan algoritma-algoritma ini sangat bergantung pada penetapan parameter kawalan.

Berdasarkan parameter ini, keputusan daripada enam kaedah yang dicadangkan dari kajian yang lalu iaitu BPSO, MSPSO, MSPSOER, BGSA, MSGSA dan MSGSAER, serta tiga kaedah yang dicadangkan dalam kajian ini berdasarkan BSKF, AMSKF, dan DESKF bagi bilangan pelaksanaan sebanyak 50 kali telah dibentangkan dalam Jadual 4.10. Rajah 4.1 menunjukkan keputusan masa pemasangan minimum (Min) yang diperolehi oleh semua kaedah yang dicadangkan dalam Jadual 4.10. Keputusan tersebut menunjukkan bahawa AMSKF dengan 1,000 lelaran dan 20 ejen berada di tempat pertama terbaik dalam mendapatkan penyelesaian optimum untuk ASP.

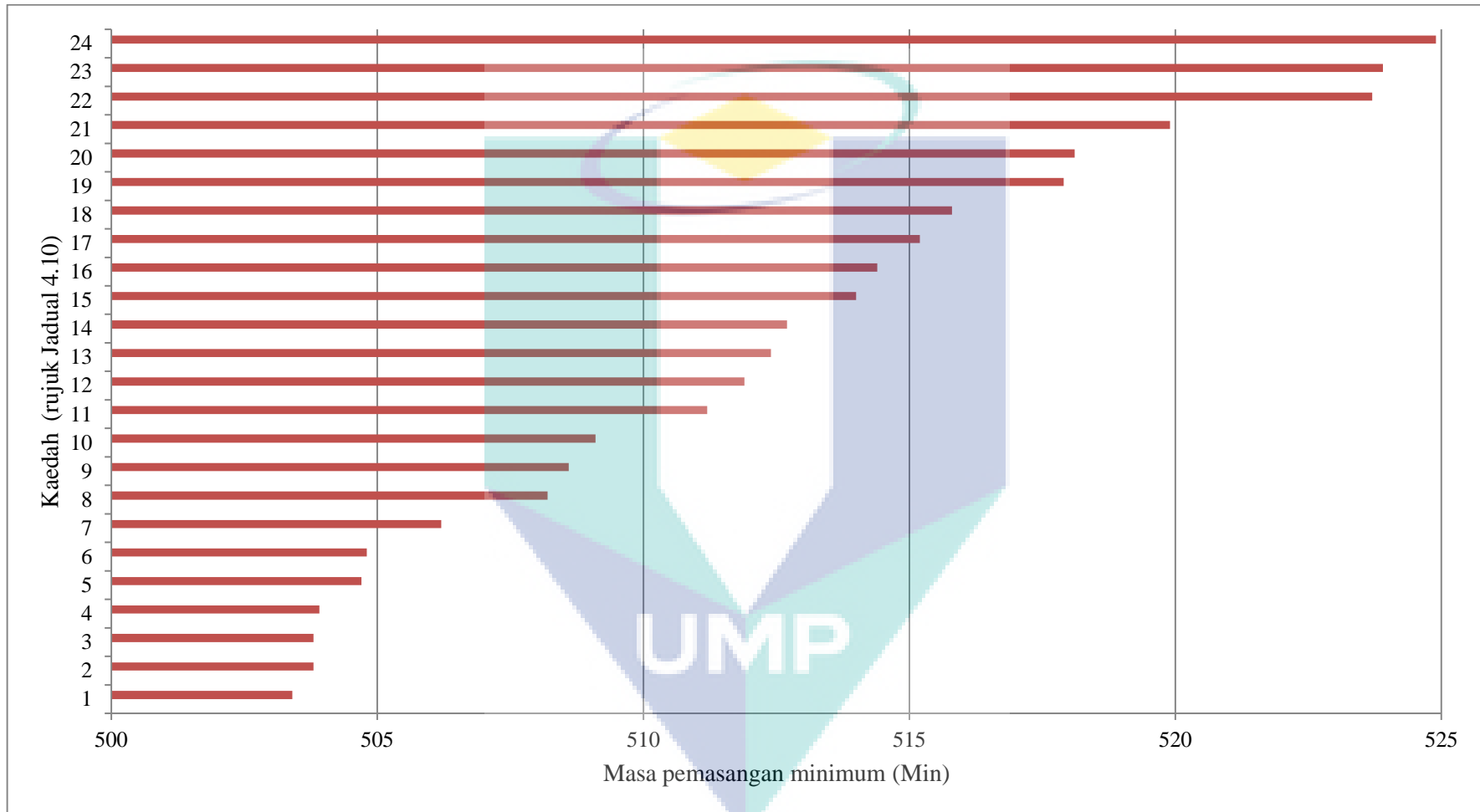
Jadual 4.9 Parameter yang digunakan untuk pendekatan yang dicadangkan berdasarkan BPSO, MSPSO, MSPSOER, BGSA, MSGSA dan MSGSAER.

Parameter	BPSO	MSPSO	MSPSOER	BGSA	MSGSA	MSGSAER
Lelaran	500	500	500	500	500	500
Jumlah populasi	50	30	30	30	30	30
Berat inerti, $\omega$	0.6	NAP	NAP	NAP	NAP	NAP
Faktor pembelajaran	1.42	NAP	NAP	NAP	NAP	NAP
$\omega$ Awal	NAP	0.9	0.9	NAP	NAP	NAP
$\omega$ Akhir	NAP	0.4	0.4	NAP	NAP	NAP
Faktor kerangka, $c1$ and $c2$	NAP	2	2	NAP	NAP	NAP
Suhu awal	NAP	NAP	NAP	100°C	NAP	NAP
Kadar penyejukan	NAP	NAP	NAP	0.95	NAP	NAP
Pemalar graviti awal, $G_o$	NAP	NAP	NAP	NAP	100	100
Pemalar, $\beta$	NAP	NAP	NAP	NAP	20	20

NAP = tidak berkenaan

Jadual 4.10 Keputusan daripada kaedah yang dicadangkan berdasarkan BPSO, MSPSO, MSPSOER, BGSA, MSGSA dan MSGSAER bersama kaedah baru yang dicadangkan berdasarkan BSKF, AMSKF, dan DESKF.

No.	Kaedah	Bilangan Lelaran	Bilangan ejen	Jumlah masa pemasangan			SD
				Minimum	Purata	Maksimum	
1	AMSKF	1,000	20	503.40	521.68	529.70	5.53
2	AMSKF	5,000	10	503.80	517.31	526.40	5.12
3	DESKF	5,000	10	503.80	518.91	524.80	4.30
4	AMSKF	1,000	30	503.91	520.19	528.70	4.94
5	AMSKF	1,000	10	504.70	524.63	534.70	5.57
6	AMSKF	5,000	20	504.80	514.39	524.20	4.98
7	AMSKF	5,000	30	506.20	513.49	522.20	3.98
8	BGSA	500	30	508.20	523.79	531.60	4.53
9	DESKF	5,000	30	508.60	515.42	519.20	2.51
10	MSGSA	500	30	509.10	519.00	525.60	3.10
11	DESKF	5,000	20	511.20	517.57	522.80	3.16
12	MSGSAER	500	30	511.90	519.00	526.40	3.20
13	DESKF	1,000	30	512.40	520.23	527.00	3.65
14	BSKF	1,000	30	512.70	535.63	549.21	7.66
15	MSPSO	500	30	514.00	530.00	538.50	4.50
16	DESKF	1,000	20	514.40	521.74	526.90	3.09
17	DESKF	1,000	10	515.20	522.68	530.70	3.57
18	BPSO	500	50	515.80	520.80	523.60	3.10
19	MSPSOER	500	30	517.90	528.60	538.40	4.30
20	BSKF	5,000	20	518.11	535.13	552.01	7.56
21	BSKF	5,000	30	519.90	533.45	544.60	5.90
22	BSKF	1,000	10	523.70	542.38	564.00	9.87
23	BSKF	5,000	10	523.90	542.40	560.71	8.06
24	BSKF	1,000	20	524.90	538.54	551.30	6.35



Rajah 4.1 Keputusan masa pemasangan minimum (Min) untuk kaedah yang dicadangkan berdasarkan BPSO, MSPSO, MSPSOER, BGSA, MSGSA dan MSGSAER, serta kaedah baru yang dicadangkan berdasarkan BSKF, AMSKF, dan DESKF.

Jadual 4.11 Keputusan mengikut turutan terbaik dan jujukan pemasangan yang berkaitan dengan pendekatan yang dicadangkan berdasarkan BPSO, MSPSO, MSPSOER, BGSA, MSGSA dan MSGSAER terhadap pendekatan yang dicadangkan berdasarkan BSKF, AMSKF, dan DESKF.

Kaedah	Bilangan lelaran	Bilangan ejen	Jumlah masa pemasangan		Jujukan pemasangan
			Minimum	Purata	
AMSKF	1,000	20	503.40	521.68	1-2-12-4-3-9-13-15-11-5-16-6-18-7-8-14-10-17-19
AMSKF	5,000	10	503.80	517.31	1-4-1-12-3-9-13-15-11-5-16-6-18-7-8-14-10-17-19
DESKF	5,000	10	503.80	518.91	2-1-12-4-3-9-13-5-15-11-16-6-18-7-8-14-10-17-19
AMSKF	1,000	30	503.91	520.19	1-3-2-4-9-12-13-15-11-5-16-6-18-7-8-14-10-17-19
AMSKF	1,000	10	504.70	524.63	2-1-12-4-3-9-13-15-11-5-16-6-18-7-8-14-10-17-19
AMSKF	5,000	20	504.80	514.39	1-2-3-4-12-9-13-15-11-5-16-6-18-7-8-14-10-17-19
AMSKF	5,000	30	506.20	513.49	1-2-12-4-3-9-13-15-11-5-16-6-7-8-18-14-10-17-19
BGSA	500	30	508.20	523.79	2-1-4-9-3-12-5-13-15-18-16-6-11-7-8-10-14-17-19
DESKF	5,000	30	508.60	515.42	1-2-12-9-3-4-13-15-11-5-16-6-18-7-8-14-10-17-19
MSGSA	500	30	509.10	519.00	2-1-4-9-12-3-5-13-15-16-18-6-11-7-8-10-14-17-19
DESKF	5,000	20	511.20	517.57	1-2-3-4-12-9-13-15-11-5-16-6-18-7-8-10-14-17-19
MSGSAER	500	30	511.90	519.00	2-15-4-1-9-12-3-5-13-18-6-16-11-7-8-14-10-17-19
DESKF	1,000	30	512.40	520.23	1-12-2-3-4-9-13-15-11-5-16-6-18-7-8-14-10-17-19
BSKF	1,000	30	512.70	535.63	2-1-12-4-3-9-13-15-11-5-6-16-18-7-8-14-10-17-19
MSPSO	500	30	514.00	530.00	2-4-3-1-9-12-5-13-15-18-16-11-6-7-8-10-14-17-19
DESKF	1,000	20	514.40	521.74	1-2-4-3-12-9-13-15-11-5-16-6-18-7-8-10-14-17-19
DESKF	1,000	10	515.20	522.68	1-2-12-4-3-9-13-15-11-5-16-6-7-8-18-14-10-17-19

Jadual 4.11 Sambungan.

Kaedah	Bilangan lelaran	Bilangan ejen	Jumlah masa pemasangan		Jujukan pemasangan
			Minimum	Purata	
BPSO	500	50	515.80	520.80	13-2-3-5-12-15-16-4-1-11-9-18-6-7-8-10-14-17-19
MSPSO ER	500	30	517.90	528.60	2-1-3-15-4-12-5-13-18-9-6-16-11-7-8-14-10-17-19
BSKF	5,000	20	518.11	535.13	2-1-12-4-3-9-13-15-11-5-16-6-18-7-8-14-10-17-19
BSKF	5,000	30	519.90	533.45	1-4-1-12-3-9-13-15-11-5-16-6-18-7-8-14-10-17-19
BSKF	1,000	10	523.70	542.38	1-3-2-4-9-12-13-15-11-5-16-6-18-7-8-14-10-17-19
BSKF	5,000	10	523.90	542.40	2-1-12-4-3-9-13-15-11-5-16-6-18-7-8-14-10-17-19
BSKF	1,000	20	524.90	538.54	2-1-12-4-3-9-13-5-11-15-16-6-18-7-8-14-10-17-19

Jadual 4.11 menunjukkan keputusan terbaik dan jujukan pemasangan yang berkaitan dengan kaedah yang dicadangkan berdasarkan BPSO, MSPSO, MSPSOER, BGSA, MSGSA dan MSGSAER bersama kaedah baru yang dicadangkan dalam kajian ini berdasarkan BSKF, AMSKF, dan DESKF. Jadual 4.11 menunjukkan bahawa AMSKF memperoleh keputusan terbaik iaitu jumlah masa pemasangan yang paling minimum untuk Minimum dan Purata berbanding dengan kaedah-kaedah lain yang dipertimbangkan bersama iaitu BSKF, DESKF, BPSO, MSPSO, MSPSOER, BGSA, MSGSA dan MSGSAER.

#### 4.4 Ringkasan

Ubjektif bab ini adalah untuk merumuskan hasil eksperimen yang diperolehi daripada kajian terdahulu dan kajian ini. Keputusan yang dilaporkan dalam bab ini diperolehi daripada beberapa siri eksperimen yang dijalankan ke atas ASP.

Bab ini bermula dengan membincangkan hasil eksperimen tiga kaedah yang dicadangkan berdasarkan algoritma SKF lanjutan iaitu BSKF, AMSKF, dan DESKF dalam menyelesaikan masalah ASP. Seterusnya, keputusan eksperimen daripada tiga algoritma SKF lanjutan tersebut dianalisis. Penyelesaian terbaik (Minimum),



penyelesaian purata (Purata), penyelesaian terburuk (Maksimum), dan sisihan piawai (SD) dalam menyelesaikan masalah ASP direkodkan. Kemudian, keputusan tersebut dibandingkan dengan prestasi kaedah yang dicadangkan dari kajian terdahulu iaitu BPSO, MSPSO, MSPSOER, BGSA, MSGSA dan MSGSAER. Hasil eksperimen yang diperolehi menunjukkan bahawa kaedah yang dicadangkan berdasarkan AMSKF mengatasi BSKF, DESKF, dan algoritma-algoritma lain dalam menyelesaikan masalah ASP untuk meminimumkan masa pemasangan. Maka, AMSKF terbukti menjadi algoritma yang paling disyorkan untuk diaplikasikan dalam masalah ASP sebenar untuk mencari penyelesaian optimum untuk tujuan meminimumkan masa pemasangan.

Keputusan yang dilaporkan dalam tesis ini sangat bergantung pada nilai parameter yang digunakan oleh algoritma-algoritma lain iaitu BPSO, MSPSO, MSPSOER, BGSA, MSGSA dan MSGSAER yang digunakan dalam kajian-kajian terdahulu. Walau bagaimanapun, dalam kajian ini, tiada percubaan untuk mengulang semula eksperimen-eksperimen untuk BPSO, MSPSO, MSPSOER, BGSA, MSGSA dan MSGSAER. Semua keputusan adalah tulen dan diambil dari kertas kajian yang diterbitkan; yang merupakan keputusan terbaik yang pernah dihasilkan oleh algoritma-algoritma tersebut dalam menyelesaikan masalah ASP.



UMP

## BAB 5

### KESIMPULAN

#### 5.1 Kesimpulan Tesis

Dalam bab 2, ASP sebagai domain masalah yang digunakan dalam kajian ini diperkenalkan. Kemudian, objektif kekangan dan pengoptimuman ASP dijelaskan. Berdasarkan kajian literatur, terdapat lebih dari 10 algoritma metaheuristik yang telah diaplikasikan untuk ASP. Namun, SKF merupakan algoritma pengoptimuman baru dan ia belum pernah diaplikasikan untuk ASP.

Pelbagai objektif dan teknik pengkomputeran telah digunakan untuk mengoptimumkan masalah ASP. Objektif ASP yang popular adalah untuk meminimumkan pertukaran peralatan, meminimumkan perubahan arah pemasangan, meminimumkan perubahan jenis pemasangan, meminimumkan kerumitan pemasangan, meminimumkan jarak perjalanan peralatan pemasangan, meminimumkan persamaan penyambung, memaksimumkan kestabilan pemasangan, mengurangkan kekangan geometri, meminimumkan kos pemasangan, meminimumkan masa pemasangan, meminimumkan masa kitaran, memaksimumkan kelancaran beban kerja, dan meminimumkan bilangan stesen kerja. Setiap aplikasi algoritma untuk menyelesaikan masalah ASP fokus untuk mencapai samada satu objektif atau berbilang objektif. Hanya algoritma metaheuristik yang telah menjalankan eksperimen terhadap ASP untuk meminimumkan masa pemasangan sahaja yang diketengahkan untuk perbincangan dalam bab 2, iaitu PSO dan GSA.

Dalam bab 3, algoritma penapis Kalman dan penapis Kalman diselakukan telah dibincangkan. Kekuatan dan kelemahan SKF juga dijelaskan dalam bab ini. Seterusnya,

algoritma SKF lanjutan iaitu BSKF, AMSKF, dan DESKF dibentangkan satu-persatu.. Aplikasi BSKF, AMSKF, dan DESKF untuk menyelesaikan masalah diskret seperti ASP telah diterangkan. Gambaran keseluruhan tentang bagaimana kajian ini dijalankan dinyatakan dengan jelas.

Akhir sekali, prestasi algoritma SKF lanjutan iaitu BSKF, AMSKF, dan DESKF dalam menyelesaikan ASP telah dinilai dan dibincangkan dalam bab 4. Kemudian, keputusan dari kajian terdahulu untuk kaedah yang dicadangkan berdasarkan BPSO, MSPSO, MSPSOER, BGSA, MSGSA, dan MSGSAER dalam menyelesaikan masalah ASP dirujuk dan dibandingkan dengan keputusan algoritma SKF lanjutan. Keputusan eksperimen yang diperoleh menunjukkan bahawa prestasi kaedah berdasarkan AMSKF telah mengatasi BSKF, DESKF, dan semua algoritma yang dicadangkan dari kajian terdahulu sebanyak 0.95% dalam menyelesaikan ASP untuk meminimumkan masa pemasangan. Maka, AMSKF terbukti menjadi algoritma yang paling disyorkan untuk digunakan untuk masalah ASP sebenar bagi mencari penyelesaian optimum untuk meminimumkan masa pemasangan.

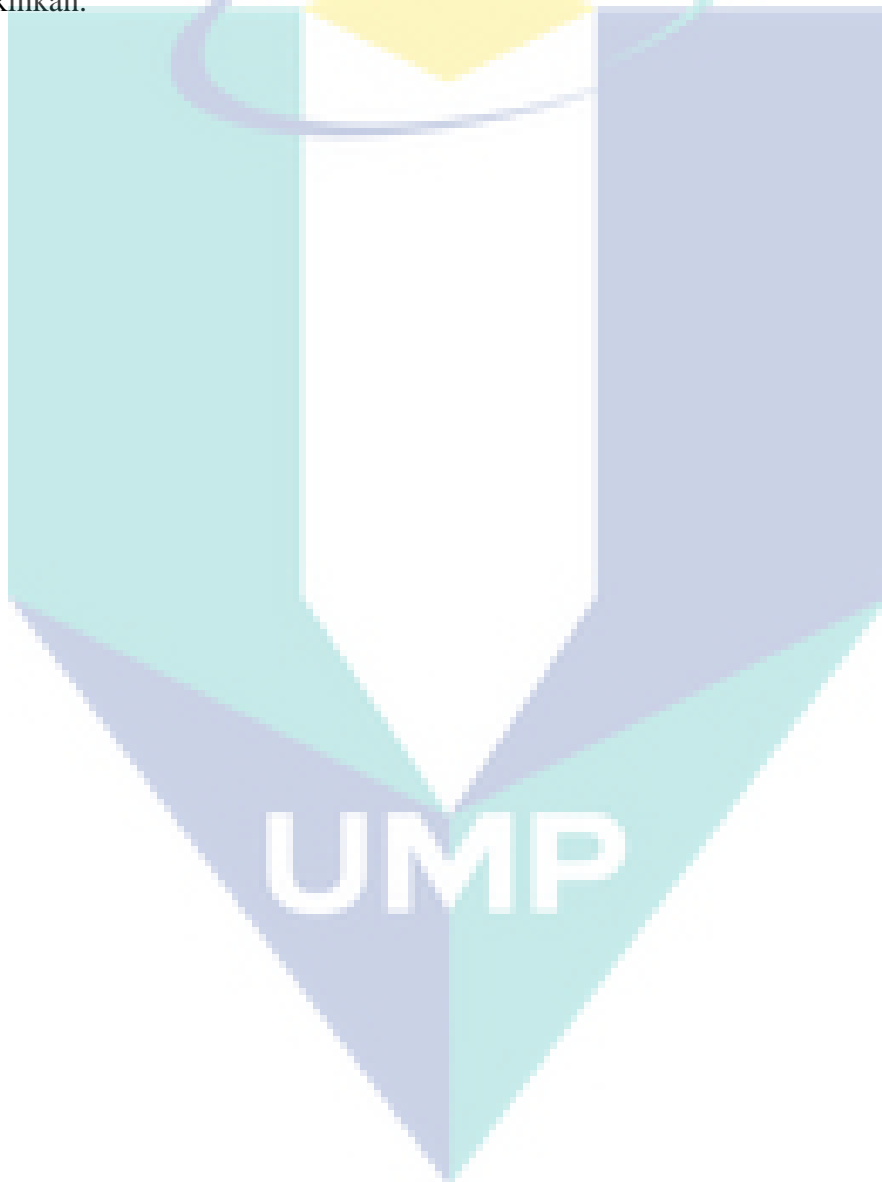
## **5.2 Sumbangan Kajian**

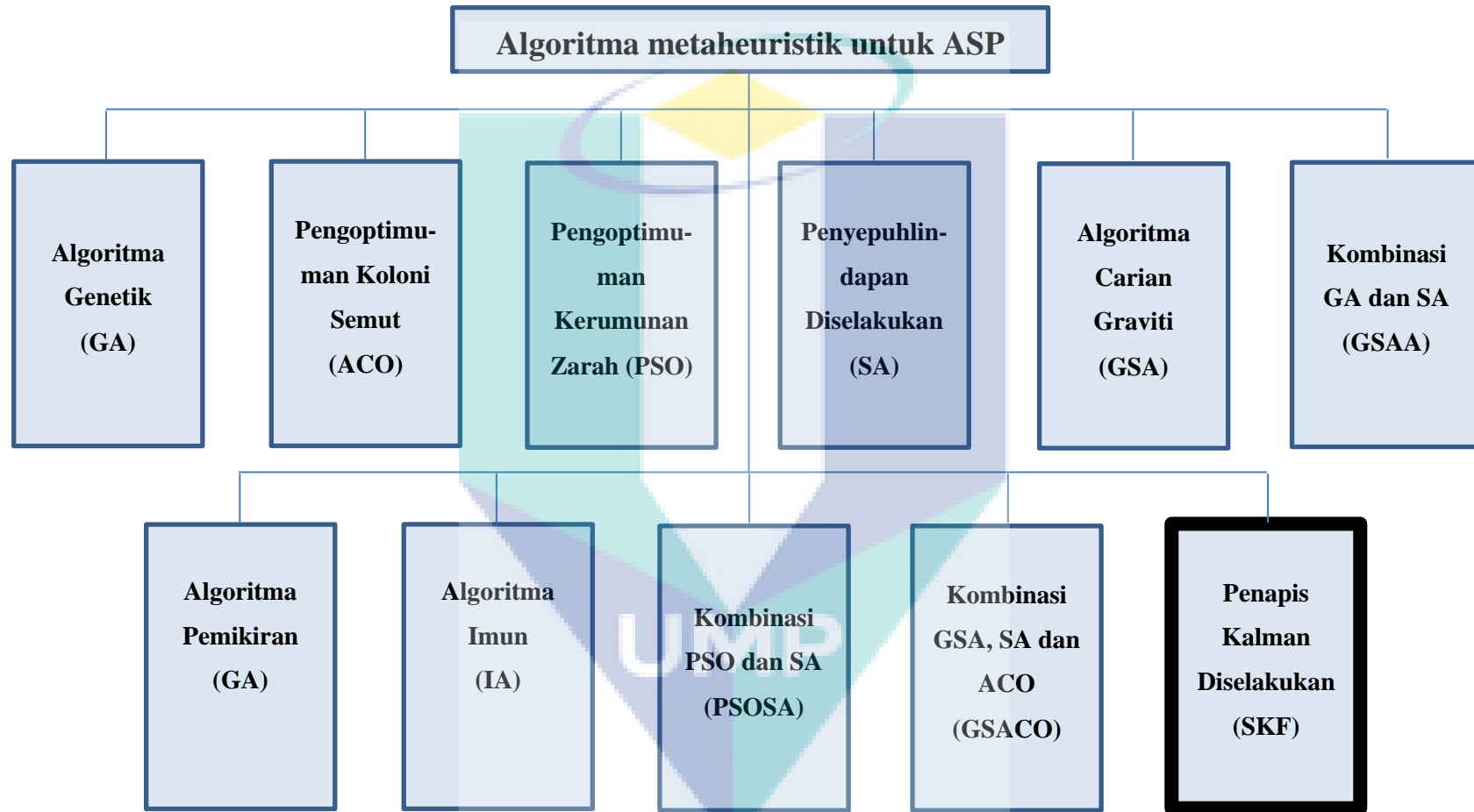
Sumbangan kajian ini adalah dalam menyelesaikan salah satu domain masalah di dalam proses pembuatan iaitu ASP menggunakan algoritma SKF lanjutan buat pertama kalinya. Kaedah yang dicadangkan berdasarkan BSKF, AMSKF, dan DESKF telah digunakan untuk mencari penyelesaian optimum bagi meminimumkan masa pemasangan. Seterusnya, ia berpotensi besar untuk mengurangkan kos pemasangan.

Kemudian, hasil eksperimen menggunakan kaedah yang dicadangkan berdasarkan BSKF, AMSKF, dan DESKF dalam menyelesaikan masalah ASP dianalisis dengan membandingkannya dengan hasil eksperimen yang diperoleh dari kajian terdahulu terhadap BPSO, MSPSO, MSPSOER, BGSA, MSGSA, dan MSGSAER untuk mengenal pasti kaedah terbaik yang paling boleh dipercayai untuk meminimumkan masa pemasangan di ASP.

### 5.3 Cadangan Masa Hadapan

Di masa hadapan, AMSKF dicadangkan untuk diaplikasikan bagi menyelesaikan masalah ASP dengan kekangan yang berbeza seperti masalah kestabilan pemasangan, tugas mesin dan stesen kerja, serta beban kerja. Selain itu, mungkin eksperimen yang dilaporkan dalam tesis ini boleh dilaksanakan semula dengan nilai parameter yang dilaraskan untuk mendapatkan hasil yang lebih baik dan perbandingan yang lebih meyakinkan.





Rajah 5.1 ASP menggunakan algoritma metaheuristik

## RUJUKAN

- Ab. Aziz, N. A., Mubin, M., Ibrahim, Z., and Nawawi, S. W. (2015). Statistical analysis for swarm intelligence — simplified. *International Journal of Future Computer and Communication*, 4(3), 193–197.
- Akaike, H. (1969). Fitting autoregressive models for prediction. *Annals of the Institute of Statistical Mathematics*, 21(1), 243–247.
- Akaike, H. (1978). A bayesian analysis of the minimum AIC procedure. *Annals of the Institute of Statistical Mathematics*, 30(1), 9–14.
- Atashpaz-Gargari, E., and Lucas, C. (2007). Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. *Proceedings of the 2007 IEEE Congress on Evolutionary Computation*, 4661–4667.
- Bai, Y. W., Chen, Z. N., Bin, H. Z., and Hun, J. (2005). An effective integration approach toward assembly sequence planning and evaluation. *Int J Adv Manuf Technol* 27(1–2), 96–105.
- Bi, X., and Wang, Y. (2011). An improved artificial bee colony algorithm. *Proceedings of the 3rd International Conference on Computer Research and Development*, 2, 174–177.
- Blum, C. and Roli, A. (2003). Metaheuristics in combinatorial optimization: overview and conceptual comparison, *ACM Computing Surveys*, 35, 268–308.
- Bonneville, F. C., Perrard, J. M. and Henrioud (1995). A genetic algorithm to generate and evaluate assembly plans, in *Proceedings of IEEE Symposium on Emerging Technologies and Factory Automation*, 2, 231-239.
- Boothroyd, G. (1994). Product design for manufacture and assembly, *Computer Aided Design*, 26, 50-59.
- Cao, P. B. and Xiao, R. B. (2007). Assembly planning using a novel immune approach. *Int J Adv Manuf Technol* 31(7–8), 770– 782.
- Chakrabarty, S. and Wolter, J. (1997). A structure-oriented approach to assembly sequence planning, *IEEE Transactions on Robotics and Automation*, 13, 14-29.
- Chang, C. C. and Tseng, H. E. and Meng, L. P. (2009). Artificial immune systems for assembly sequence planning exploration. *Eng Appl Artif Intell* 22(8), 1218–1232.
- Chen, G., Zhou, J., Cai, W., Lai, X., Lin, Z. and Menassa, R. (2006). A framework for an automotive body assembly process design system. *CAD Comput Aided Des* 38(5), 531–539.

- Chen, R. S., Lu, K. Y. and Yu, S. C. (2002). A hybrid genetic algorithm approach on berbilang-objective of assembly planning problem. *Eng Appl Artif Intell* 15(5), 447–457.
- Chen, S. F., Liu, S. F. and Liu Y. J. (2001). An adaptive genetic assembly-sequence planner. *Int J Comput Integr Manuf* 14(5), 489–500.
- Chen, W. C., Tai, P. H., Deng, W. J. and Hsieh, L. F. (2008). A three-stage integrated approach for assembly sequence planning using neural networks. *Expert Syst Appl* 34(3), 1777–1786.
- Chen, W.C., Tai, P. H., Deng, W. J. and Hsieh, L. F. (2008). A three-stage integrated approach for assembly sequence planning using neural networks, *Expert Syst Appl*, 34, 1777 – 1786.
- Cheng, S., and Shi, Y. (2011). Diversity control in particle swarm optimization. *Proceedings of the 2011 IEEE Symposium on Swarm Intelligence*, 1–9.
- Choi, Y. K., Lee, D. M. and Cho, Y. B. (2008). An approach to berbilang-criteria assembly sequence planning using genetic algorithms, *The International Journal of Advanced Manufacturing Technology*, 42, 180-188.
- Choi, Y. K., Lee, D. M., and Cho, Y. B. (2009). An approach to berbilang-criteria assembly sequence planning using genetic algo-rithms. *Int J Adv Manuf Technol* 42(1–2), 180–188.
- De, L. P., Latinne, P. and Rekiek, B. (2001). Assembly planning with an ordering genetic algorithm, *Int J Prod Res*, 39, 3623 – 3640.
- DeLit, P., Latinne, P., Rekiek, B. and Delchambre, A. (2001). As-sembly planning with an ordering genetic algorithm. *Int J Prod Res* 39(16), 3623–3640.
- Derbel, F. (2001). Modeling fire detector signals by means of system identification techniques. *IEEE Transactions on Instrumentation and Measurement*, 50(6), 1815–1821.
- Derrac, J., García, S., Molina, D., and Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1), 3–18.
- Dorigo, M., Maniezzo, V., and Coloni, A. (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 26(1), 29–41.
- Eusuff, M., Lansey, K., and Pasha, F. (2006). Shuffled frog-leaping algorithm: A memetic meta-heuristic for discrete optimization. *Engineering Optimization*, 38(2), 129-154.

- Farmer, J. D., Packard, N. H., and Perelson, A. S. (1986). The immune system, adaptation, and machine learning. *Physica D: Nonlinear Phenomena*, 22(1–3), 187-204.
- Gao, L., Qian, W. R., Li, X. Y. and Wang, J. F. (2010). Application of memetic algorithm in assembly sequence planning, *Int J Adv Manuf Technol*, 49, 1175-1184.
- Gao, L., Qian, W., Li, X. and Wang, J. (2010). Application of memetic algorithm in assembly sequence planning. *Int J Adv Manuf Technol* 49(9–12), 1175–1184.
- Gao, X. Z., Wang, X., and Ovaska, S. J. (2010). A hybrid harmony search method based on OBL. *Proceedings of the 13th IEEE International Conference on Computational Science and Engineering*, 140–145.
- Glover, F. (1989). Tabu search - Part I. *ORSA Journal on Computing*, 1(3), 190–206.
- Guan, Q., Liu, J. H. and Zhong, Y. F. (2002). A concurrent hierarchical evolution approach to assembly process planning. *Int J Prod Res* 40(14), 3357–3374.
- Guo, Y. W., Li, W. D., Mileham, A. R. and Owen, G. W. (2009). Applications of particle swarm optimization in integrated process planning and scheduling, *Robot Comput Integr Manuf*, 25, 280-288.
- Holland, J. H. (1975). *Adaptation in natural and artificial systems : An introductory analysis with applications to biology, control, and artificial intelligence*. Ann Arbor : University of Michigan Press.
- Homem de Mello, L. S. and Arthur, C. D. (1990). AND/OR graph representation of assembly plans, *IEEE Transactions on Robotics and Automation*, 6, 188-199.
- Hong, D. S. and Cho, H. S. (1995). A neural network based computational scheme for generating optimized robotic assembly sequences, *Engineering Application Artificial Intelligence*, 8, 129-145.
- Huang, H. H., Wang, M. H. and Johnson, M. R. (2000). Disassembly sequence generation using a neural network approach, *Journal of Manufacturing Systems*, 19, 73-82.
- Hui, C., Yuan. L. and Kai-Fu, Z. (2009). Efficient method of assembly sequence planning based on GAAA and optimizing by assembly path feedback for complex product. *Int J Adv Manuf Technol* 42(11–12), 1187–1204.
- Ibrahim, I., Ibrahim, Z., Ahmad, H., Md Yusof, Z., Nawawi, S. W., Mohamad, M. S., Mubin, M. and Mokhtar, N. (2014). An Assembly Sequence Planning Approach with Binary Gravitational Search Algorithm, *Frontiers in Artificial Intelligence and Applications*, 15, 179-193.



- Ibrahim, Z., Abd Aziz, N. H., Ab Aziz, N. A., Razali, S., Shapiai, M. I., Nawawi, S. W., and Mohamad, M. S. (2015). A Kalman filter approach for solving unimodal optimization problems. *ICIC Express Letters*, 9(12), 3415–3422.
- Ibrahim, Z., Nawawi, S. W., Razali, S., Muhammad, B., Mohd Azmi, K. Z., Aspar, Z., and Ab Aziz, N. A. (2015). Simultaneous computation of model order and parameter estimation of a heating system based on particle swarm optimization for autoregressive with exogenous model. *ICIC Express Letters*, 9(4), 1159–1165.
- Julai, S., Tokhi, M. O., Mohamad, M., and Latiff, I. A. (2010). Active vibration control of a flexible plate structure using particle swarm optimization. *Proceedings of the 9th IEEE International Conference on Cybernetic Intelligent Systems*, 1–6.
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1), 35–45.
- Karaboga, D., and Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39(3), 459–471.
- Kaveh, A., and Farhoudi, N. (2013). A new optimization method: Dolphin echolocation. *Advances in Engineering Software*, 59, 53–70.
- Kaveh, A., and Talatahari, S. (2010). A novel heuristic optimization method: Charged system search. *Acta Mechanica*, 213(3), 267–289.
- Kennedy, J., and Eberhart, R. (1995). Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks*, 4, 1942–1948.
- Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671–680.
- Laperriere, L. and El Marghy, A. (1994). Assembly sequence planning for simultaneous engineering. *International Journal of Advance Manufacturing Technology*, 9, 231-244.
- Lazzerini, B. and Marcelloni, F. (2000). Genetic algorithm for generating optimal assembly plans. *Artif Intell Eng* 14(4), 319– 329.
- Lee, S. and Shin, Y. G. (1990). Assembly planning based on geometric reasoning. *Computation and Graphics*, 14, 237-250.
- Li, J. R., Khoo, L. P., Tor, S. B. (2003). A Tabu-enhanced genetic algorithm approach for assembly process planning. *J Intell Manuf* 14(2), 197–208.
- Li, M., Wu, B., Hu, Y., Jin, C. and Shi, T. (2013). A hybrid assembly sequence planning approach based on discrete particle swarm optimization and evolutionary direction operation, *The International Journal of Advanced Manufacturing Technology*, 68, 617-630.

- Li, S. X. and Shan, H. B. (2008). GSSA and ACO for assembly sequence planning: a comparative study. In: Proceedings of the IEEE international conference on automation and lo-gistics, ICAL 2008, 1270–1275.
- Liang, J. J., Qu, B. Y., and Suganthan, P. N. (2014). Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. Technical Report 201311, Computational Intelligence Laboratory, Zhengzhou University, China And Technical Report, Nanyang Technological University, Singapore.
- Liu, Z., Hansson, A., and Vandenberghe, L. (2013). Nuclear norm system identification with missing inputs and outputs. *Systems and Control Letters*, 62(8), 605–612.
- Ljung, L. (1987). *System identification : Theory for the user*. USA: Prentice-Hall.
- Lu, C., Wong, Y. S. and Fuh, J. Y. H. (2005). An enhanced assembly planning approach using a berbilang-objective genetic algorithm. *Proc Inst Mech Eng, B J Eng Manuf* 220(2), 255–272.
- Lu, C., Wong, Y. S., and Fuh, J. Y. H. (2006). An enhanced assembly planning approach using a berbilang-objective genetic algorithm, *Proc Inst Mech Eng Part B J Eng Manuf*, 220, 255-272.
- Lv, H. and Lu, C. (2010). An assembly sequence planning approach with a discrete particle swarm optimization algorithm. *Int J Adv Manuf Technol* 50(5–8), 761.
- Lv, H. G. and Lu, C. (2010) An assembly sequence planning approach with a discrete particle swarm optimization algorithm, *The International Journal of Advanced Manufacturing Technology*, 50, 761-770.
- Lv, H. G., Lu, C. and Zha, J. (2010) A hybrid DPSO-SA approach to assembly sequence planning. In: *IEEE international conference on mechatronics and automation, ICMA 2010*, 5589203, 1998–2003.
- Mahmoud, M. S., and Qureshi, A. (2012). Model identification and analysis of small-power wind turbines. *International Journal of Modelling, Identification and Control*, 17(1), 19–31.
- Marian, R. M., Luong, L. H. S. and Abhary, K. (2003) Assembly sequence planning and optimization using genetic algorithms: part i. automatic generation of feasible assembly sequences, *Applied Soft Computing*, 2, 223-253.
- Marian, R. M., Luong, L. H. S. and Abhary, K. (2006) A genetic algorithm for the optimisation of assembly sequences. *Comp Ind Eng* 50(4), 503–527.
- Md Yusof, Z., Ibrahim, I., Ibrahim, Z., Mohd Azmi, K. Z., Abd Aziz, N. H., Ab Aziz, N. A., and Mohamad, M. S. (2016). AMSKF : Angle modulated simulated Kalman filter algorithm for combinatorial optimization problems. *ARPN Journal of Engineering and Applied Sceinces*, 4854-4859.

- Md Yusof, Z., Ibrahim, I., Ibrahim, Z., Mohd Azmi, K. Z., Abd Aziz, N. H., Ab Aziz, N. A., and Mohamad, M. S. (2016). DESKF : Distance evaluated simulated Kalman filter algorithm for combinatorial optimization problems. *ARNP Journal of Engineering and Applied Sciences*, 4911-4916.
- Md Yusof, Z., Ibrahim, I., Satiman, S. N., Ibrahim, Z., Abd Aziz, N. H., and Ab Aziz, N. A. (2015). BSKF : Binary simulated Kalman filter. *Third International Conference on Artificial Intelligence, Modelling and Simulation*, 77–81.
- Milner, J. M., Graves, S. C. and Whitney, D. E. (1994) Using simulated annealing to select least-cost assembly sequences, in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2058 – 2063.
- Mirjalili, S. (2015). The ant lion optimizer. *Advances in Engineering Software*, 83, 80–98.
- Mitrovic-Minic, S. and Krishnamurti, V. (2006) The berbilingple TSP with time windows: Vehicle bounds based on precedence graphs. *Oper Res Lett* 34(1), 111–120.
- Mohd Azmi, K. Z. (2017). Opposition-based simulated Kalman filters and their application in system identification, *Universiti Malaysia Pahang, Malaysia*.
- Mohd Azmi, K. Z., Ibrahim, Z., Pebrianti, D., Nawawi, S. W., and Ab Aziz, N. A. (2015). Simultaneous computation of model order and parameter estimation of a heating system based on gravitational search algorithm for autoregressive with exogenous inputs. *ARNP Journal of Engineering and Applied Sciences*, 10(2), 633–642.
- Mohd Azmi, K. Z., Pebrianti, D., Ibrahim, Z., Sudin, S., and Nawawi, S. W. (2015). Simultaneous computation of model order and parameter estimation for system identification based on gravitational search algorithm. *6th International Conference on Intelligent Systems, Modelling and Simulation*, 135–140.
- Moor, B. De, Gersem, P. De, Schutter, B. De, and Favoreel, W. (1997). DAISY : A database for identification of systems. *Journal A*, 38(3), 4–5.
- Motavalli, S. and Islam, A. (1997) Berbilang-criteria assembly sequencing, *Computational Industrial Eng.*, 32 (1997), 743–751.
- Muhammad, B., Ibrahim, Z., Ghazali, K. H., Mohd Azmi, K. Z., Ab Aziz, N. A., Abd Aziz, N. H., and Mohamad, M. S. (2015). A new hybrid simulated Kalman filter and particle swarm optimization for continuous numerical optimization problems. *ARNP Journal of Engineering and Applied Sciences*, 10(22), 17171–17176.
- Mukred, J. A. A., Ibrahim, Z., Ibrahim, I., Adam, A., Wan, K., Md. Yusof, Z. and Mokhtar, N. (2012) A binary particle swarm optimization approach to optimize assembly sequence planning, *Advanced Science Letters*, 13, 732-738.

- Mustapa, A., Ibrahim, Z., Md Yusof, Z. and Ibrahim, I. (2016) Global Optimization Distance Evaluated Particle Swarm Optimization for Combinatorial Optimization Problem. The National Conference for Postgraduate Research (NCON-PGR).
- Mustapa, A., Md. Yusof, Z., Adam, A., Muhammad, B. and Ibrahim, Z. (2017) Solving Assembly Sequence Planning using Angle Modulated Simulated Kalman Filter. The 4<sup>th</sup> Asia Pacific Conference on Manufacturing Systems and The 3<sup>rd</sup> International Manufacturing Engineering Conference (APCOMS-IMEC).
- Pan, C., Smith, S. and Smith, G. (2006) Automatic assembly sequence planning from STEP CAD files. *Int J Comput Integr Manuf* 19(8), 775–783.
- Passino, K. M. (2002). Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine*, 22(3), 52–67.
- Pham, D. T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., and Zaidi, M. (2006). The bees algorithm—A novel tool for complex optimization. *Proceedings of the 2nd International Virtual Conference on Intelligent Production Machines and Systems*, 454–459.
- Qin, Y. F. and Xu, Z. G. (2007) Assembly process planning using a berbilang-objective optimization method. In: *Proceedings of the 2007 IEEE international conference on mechatronics and automation, ICMA 2007*, 4303610, 593–598.
- Rashedi, E., Nezamabadi-pour, H., and Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Information Sciences*, 179(13), 2232–2248.
- Senin, N., Groppetti, R. and Wallace, D. R. (2000) Concurrent assembly planning with genetic algorithms. *Robot Comput Integr Manuf* 16(1), 65–72.
- Shah-Hosseini, H. (2007) Problem solving by intelligent water drops. *Proceedings of the 2007 IEEE Congress on Evolutionary Computation*, 3226–3231.
- Shan, H., Li, S., Gong D, Lou P (2006) Genetic simulated annealing algorithm-based assembly sequence planning. In: *IET conference publications*, vol 524, 1573–1579.
- Shan, H., Zhou, S., Sun, Z. (2009) Research on assembly sequence planning based on genetic simulated annealing algo-rithm and ant colony optimization algorithm. *Assem Autom* 29(3), 249–256.
- Shuang, B., Chen, J., Li, Z. (2008) Microrobot based micro-assembly sequence planning with hybrid ant colony algorithm. *Int J Adv Manuf Technol* 38(11–12), 1227–1235.
- Siddique, N., and Adeli, H. (2015) Nature inspired computing: An overview and some future directions. *Cognitive Computation*, 7(6), 706–714.

- Simon, D. (2008) Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation*, 12(6), 702–713.
- Sinanoglu, C. and Boklu, H. R. (2005) An assembly sequence planning system for mechanical parts using neural network. *Assem Autom* 25(1), 38–52.
- Smith, G. C. and Smith, S. S. F. (2002) An enhanced genetic algorithm for automated assembly planning. *Robot Comput-Integr Manuf* 18(5–6), 355–364.
- Smith, S. S. F. (2004) Using berbilang genetic operators to reduce premature convergence in genetic assembly planning. *Comput Ind* 54(1), 35–49.
- Smith, S. S.F. and Liu, Y. J. (2001) The application of berbilang-level genetic algorithms in assembly planning. *J Ind Technol* 17(4), 1.
- Su, Q. (2009) A hierarchical approach on assembly sequence planning and optimal sequences analyzing, *Robotics and Computer-Integrated Manufacturing*, 25, 224–234.
- Su, Q. (2009) A hierarchical approach on assembly sequence planning and optimal sequences analyzing. *Robot Comput-Integr Manuf* 25(1), 224–234.
- Talbi, E. G. (2009) *Metaheuristics: From Design to Implementation*, John Wiley & Sons.
- Tseng, H. E. and Tang, C. E. (2006) A sequential consideration for assembly sequence planning and assembly line balancing using the connector concept. *Int J Prod Res* 44(1), 97–116.
- Tseng, H. E., Chen, M. H., Chang, C. C. and Wang, W. P. (2008) Hybrid evolutionary berbilang-objective algorithms for integrating assembly sequence planning and assembly line balancing. *Int J Prod Res* 46(21), 5951–5977.
- Tseng, H. E., Li, J. D. and Chang, Y. H. (2004) Connector-based approach to assembly planning using a genetic algorithm. *Int J Prod Res* 42(11), 2243–2261.
- Tseng, H. E., Wang, W. P. and Shih, H. Y. (2007) Using memetic algorithms with guided local search to solve assembly sequence planning. *Expert Syst Appl* 33(2), 451–467.
- Tseng, Y. J., Chen, J. Y. and Huang, F. Y. (2010) A berbilang-plant assembly sequence planning model with integrated assembly sequence planning and plant assignment using GA. *Int J Adv Manuf Technol* 48(1–4), 333–345.
- Tseng, Y. J., Chen, J. Y. and Huang, F. Y. (2010) A particle swarm optimisation algorithm for berbilang-plant assembly sequence planning with integrated assembly sequence planning and plant assignment. *Int J Prod Res* 48(10), 2765–2791.

- Tseng, Y. J., Yu, F. Y. and Huang, F. Y. (2011) A green assembly sequence planning model with a closed-loop assembly and disassembly sequence planning using a particle swarm optimization method, *Int J Adv Manuf Technol*, 57, 1183 – 1197.
- Tseng, Y. J., Yu, F.Y and Huang, F. Y. (2010) A berbilang-plant assembly sequence planning model with integrated assembly sequence planning and plant assignment using GA, *Int J Adv Manuf Technol*, 48, 333 – 345.
- Tutunji, T., Saleem, A., Salah, M., and Ahmad, N. (2013). Identification of piezoelectric ultrasonic transducers for machining processes. *Proceedings of the 9th International Symposium on Mechatronics and its Applications*, 9–13.
- Udeshi, T. and Tsui, K. (2005) Assembly sequence planning for automated micro assembly. In: *IEEE International symposium on assembly and task planning 2005*, vol 2005, 98– 105.
- Wang, H., Wu, Z., Liu, Y., Wang, J., Jiang, D., and Chen, L. (2009). Space transformation search: a new evolutionary technique. *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, 537-544.
- Wang, J. F., Liu, J. H. and Zhong, Y. F. (2005) A novel ant colony algorithm for assembly sequence planning, *Int J Adv Manuf Technol*, 25, 1137-1143.
- Wang, J. F., Liu, J. H. and Zhong, Y. F. (2005) A novel ant colony algorithm for assembly sequence planning. *Int J Adv Manuf Technol* 25(11–12), 1137-1143.
- Wang, W. P., and Tseng, H. E. (2009) Complexity estimation for genetic assembly sequence planning. *J Chin Inst Ind Eng* 26(1), 44–52.
- Wang, Y. and Liu, J. H. (2010) Chaotic particle swarm optimization for assembly sequence planning. *Robot Comput Integr Manuf* 26(2), 212–222.
- Welch, G., and Bishop, G. (1995). An introduction to the Kalman filter. *Technical Report 95-041*. University of North Carolina at Chapel Hill.
- Xiaoming, Z. and Pingan, D. (2008) A model-based approach to assembly sequence planning, *International Journal of Advanced Manufacturing Technology*, 39, 983–994.
- Yang, X. S. (2009). Firefly algorithms for berbilangmodal optimization. *Proceedings of the 5th International Symposium on Stochastic Algorithms: Foundations and Applications*, *Lecture Notes in Computer Science*, 5792, 169–178.
- Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, 284, 65–74.
- Yang, X. S. (2012). Flower pollination algorithm for global optimization. *Unconventional Computation and Natural Computation*, 7445, 240–249.

- Yang, X. S., and Deb, S. (2009). Cuckoo search via Levy flights. Proceedings of the 2009 World Congress on Nature and Biologically Inspired Computing, 210–214.
- Yu, H., Yu, J. and Zhang, W. (2009) An particle swarm optimization approach for assembly sequence planning. Appl Mech Mater 1228, 16–19.
- Zha, X.F. (2000) An object-oriented knowledge based petri net approach to intelligent integration of design and assembly planning, Artificial Intelligence in Engineering, 14, 83-112.
- Zhang, J., Sun, J. and He, Q. (2010) An approach to assembly sequence planning using ant colony optimization. In: Proceedings of 2010 international conference on intelligent control and information processing, ICICIP 2010, vol part 2, 230–233.
- Zhang, W. (1989) Representation of assembly and automatic robot planning by petri net, IEEE Transactions on Systems, Man, and Cybernetics, 19, 418-422.
- Zhao, P. J. (2010). A hybrid harmony search algorithm for numerical optimization. Proceedings of the 2010 International Conference on Computational Aspects of Social Networks, 255–258.
- Zhou, W., Zheng, J. R., Yan, J. J. and Wang, J. F. (2011) A novel hybrid algorithm for assembly sequence planning combining bacterial chemotaxis with genetic algorithm, Int J Adv Manuf Technol, 715 – 724.
- Zhou, W., Zheng. J., Yan, J. and Wang, J. (2010) A novel hybrid algorithm for assembly sequence planning combining bac-terial chemotaxis with genetic algorithm. Int J Adv Manuf Technol 52(5–8), 715–724.

The logo for UMP (University of Management and Production) is a large, stylized letter 'V' shape. The top part of the 'V' is a light blue triangle pointing upwards. The bottom part of the 'V' is a light blue triangle pointing downwards. The two triangles meet at a point in the center. The letters 'UMP' are written in a bold, white, sans-serif font across the center of the 'V' shape.

UMP

**LAMPIRAN A**  
**SENARAI PENERBITAN**

Persidangan

1. Ainizar Mustapa, Zuwairie Ibrahim, Zulkifli Md Yusof, and Ismail Ibrahim, “Global Optimization Distance Evaluated Particle Swarm Optimization for Combinatorial Optimization Problem”, The National Conference for Postgraduate Research (NCON-PGR), 2016.
2. Ainizar Mustapa, Zulkifli Md. Yusof, Asrul Adam, Badaruddin Muhammad, and Zuwairie Ibrahim, “Solving Assembly Sequence Planning using Angle Modulated Simulated Kalman Filter”, The 4<sup>th</sup> Asia Pacific Conference on Manufacturing Systems and The 3<sup>rd</sup> International Manufacturing Engineering Conference (APCOMS-IMEC), 2017.



UMP