Chapter 2

Literature Review

The review of the previous literatures related to the studies in this dissertation is presented in this chapter. The review contexts can be classified into four main categories. Firstly, the review will focus on a *Single Machine Scheduling* (SMS) research. The single machine scheduling research can be further divided into three parts according to their objective functions, which are sequence-dependent setup problem, earliness/tardiness problems, and tardiness with sequence-dependent setup time problem. Secondly, the previous research on *Vehicle Routing Problem with Time Windows* (VRPTW) problem will be described. A home care worker scheduling research as well as brief review of staff planning and scheduling problem will be presented in the third section. Finally, the theory and applications of the proposed solution techniques, i.e. *Branch and Bound* (B&B) algorithm and *Particle Swarm Optimization* (PSO) algorithm, are presented.

2.1 Single Machine Scheduling Research

The single machine scheduling problem has long been studied by many researchers. In order to get broad ideas about early single machine scheduling research, Gupta and Kyparisis (1987) provided an overview of early static single machine scheduling research conducted from the 1950s through the mid 1980s. The review of related literatures of this article is shown as follows.

2.1.1 Sequence-dependent Setup (changeover) Problem

Many solution techniques have been proposed for the single machine problem with sequence dependent setup (changeover) cost and time. Glassey (1968) proposed an algorithm to determine the job sequence that minimizes the total number of changeovers of the machine when tardiness is not allowed. The network model is constructed and the dynamic programming formulation is used to obtain the shortest route that corresponds to the minimum changeovers. The similar work but with the objective of minimizing sequence-dependent changeover penalty is considered by Driscoll and Emmons (1977). They reported the use of forward-time and backward-time dynamic programming with the application of monotonicity property to find an optimal solution. Gascon and Leachman (1988) also developed an approach similar to that of Glassey for production scheduling of multiple items with time-varying deterministic demands when idle time is allowed. The objective is to minimize the sum of changeover and inventory holding costs by using dynamic programming algorithm to find the optimal path of the shortest route problem. A more specific problem of setup cost was considered by Hu *et al*. (1987). The setup cost has special structure. The setup cost when the production line changes from producing item *i* to item *j* is one dollar if *i* is less than *j*, and zero if *i* is greater than or equal to *j*. The setup time of all studies is assumed to be negligible.

The sequence-dependent setup time problem is usually modeled as a *Traveling Salesman Problem* (TSP). Gavett (1965) and Haynes *et al*. (1973) provided several heuristics to find a near optimal solution for the sequence-dependent setup time problem. White and Wilson (1977) investigated a procedure that classified sequence-dependent setup operations and predicts the setup times and then develop a quantitative heuristic scheme for sequencing a job with the objective of minimizing total setup time. The problem is a variation of TSP problem. Lockett and Muhlemann (1972) presented a branch and bound algorithm and heuristics to minimize the number of tools changes on a single machine with multiple tools. Since, each tool change takes the same amount of time, the objective is equivalent to minimize total setup time. Liao and Yu (1996) developed a batching and sequencing scheme that minimize the makespan (which is equivalent to minimize the total setup time) when tardiness is not allowed. The two sequencing heuristics are proposed to generate a near optimal solution for the problem. For comprehensive survey of the literature on problem involving setup considerations, readers are referred to the paper by Allahverdi *et al*. (1999).

2.1.2 Earliness and Tardiness Problem

The scheduling problems with earliness and tardiness penalties have received much interest due to the growing adoption of the JIT manufacturing philosophy. A special case known as a common due date problem was studied by a number of researchers. Mondal and Sen (2001) investigated a graph search space algorithm with depth-first Branch and Bound scheme to the weighted earliness and tardiness problem with a restricted (small) due date. Azizoglu and Webster (1997) introduced a Branch and Bound algorithm and a Beam Search procedure to solve the problem with the sequence-dependent family setup time where the due date is common and unrestricted (large). A recent paper by Rabadi *et al*. (2004) considered the same problem with sequence-dependent setup time but earliness and tardiness are weighted equally. The Branch and Bound algorithm was developed to solve the problem with up to 25 jobs.

The scheduling problem with non-identical due dates was investigated by many researchers. Potts and Wassenhove (1985) presented a Branch and Bound algorithm with Lagrangian relaxation and multiplier adjustment procedure for the total weighted tardiness problem. Fatih *et al*. (2004) applied PSO algorithm to the same problem. The PSO can give the near-optimal solution but can solve larger size problem. For a weighted earliness and tardiness case, Abdul-Razaq and Potts (1988) proposed a Branch and Bound technique with the use of a relaxed Dynamic Programming procedure by mapping state-space onto a smaller state-space and performing recursion to obtain a good lower bound. The lower bound is further improved through the use of state-space modifier. The problem can handle up to 20 jobs with reasonable computational time. Ow and Morton (1989) presented a series of heuristics, dispatch priority rules and a filtered beam search technique, to provide near optimal solution with relatively small search tree when the number of jobs is less than 30. Ibaraki and Nakamura (1994) used a Successive Sublimation Dynamic Programming method for the same problem by performing several types of dynamic programming recursions on a number of generated states. The method succeeds in effectively solving problems with up to 35 jobs. Li (1997) and Liaw (1999) developed a Branch and Bound algorithm for same problem but with different elimination criteria and lower bounding procedure. The computational experiments show that both techniques perform very well on problems with up to 50 jobs. Coleman (1992) proposed a simple mixed Integer Programming model for a weighted earliness and tardiness problem with sequence-dependent setup time. The model can handle the problem up to 8 jobs. Yano and Kim (1991) discussed a problem where the earliness and tardiness weights are proportional to the processing times of the jobs.

Dominance criteria are derived which eliminate many possible sequences from consideration in the Branch and Bound procedure. Many authors considered the problem where the idle time can be inserted into the schedule. Chang (1999) applied Branch and Bound algorithm with the lower bound based on Overlap Elimination procedure for the un-weighted earliness and tardiness problem. Ventura and Radhakrishnan (2003) formulated the same problem as 0-1 linear program. They implement Lagrangian relaxation procedure that utilizes the sub-gradient algorithm to obtain near optimal solutions. Shaller (2004) presented a timetabling algorithm that inserts idle time into a schedule in order to minimize the sum of job earliness and the square of job tardiness and establish a lower bound for a Branch and Bound method. The weighted earliness and tardiness case was considered by Chen and Lin (2002). They utilized a dominance rule to develop a relationship matrix and combined this matrix with a branching strategy to solve the problem. The most recent paper by Sourd (2005) also used Branch and Bound procedure for the same problem but with sequence-dependent setup time and cost between groups of products. The computational result has shown that the algorithm is limited to problems with no more than 20 jobs. Those interested in early/tardy problems are referred to Baker and Scudder (1990) for comprehensive survey of the previous literature.

Since the problem of Sourd (2005) is general and complex, the representation of the schedule and the algorithm are complicated. This results in a long computational time. Hence, it is only applicable to small-sized problems (up to 20 jobs). The first focused topic of this paper aims to develop a simpler algorithm and schedule representation, which also guarantees the global optimal solution and can be used to solve larger problem. However, some assumptions are required. The problem under consideration in this paper is based on assumptions that the idle time cannot be inserted into the schedule and setup time is sequence-independent. In this case, the setup time can be added directly to the processing time. The insertion of idle time in the schedule can reduce the earliness of the jobs. However, this may not be appropriate for some situations, where the idle cost is higher than the earliness penalties, the capacity of machine is less than the demand, or the machine under consideration is a bottleneck resource, etc.

2.1.3 Total Tardiness Problem with Sequence-Dependent Setup Time

The single machine total tardiness problem with sequence-dependent setup time has been investigated by many researchers and many problem-solving procedures have been proposed. Rubin and Ragatz (1995) developed a *Genetic Algorithm* (GA) for this problem. They compared the performance of *Genetic Algorithm* to *Branch and Bound* (B&B) algorithm and *Random Start Pairwise Interchange* (RS) and concluded that RS and GA both techniques outperformed B&B in terms of solution quality and computational time. B&B can be used to find an optimal solution for small-sized problem. For large-sized problem, with the limited number of nodes, B&B returned sub-optimal schedule and cannot reach as good solution quality as RS and GA. Despite its simplicity, RS outperformed GA in many instances.

Tan and Narasimhan (1997) proposed the *Simulated Annealing* (SA) technique to solve the same problem. They used RS technique as a benchmark for conducting a comparison with SA. The experimental results indicated that the computational time of SA is slightly higher than RS. But the slightly increase in computational time is offset by better solution quality, i.e. lower tardiness value.

The comparative performance of four techniques above, i.e., Branch and Bound, Genetic Algorithm, Random Start Pairwise Interchange, Simulated Annealing was evaluated by Tan *et al*. (2000). The direct comparison on the various set of test problems by Rubin and

Ragatz (1995) were performed. The test data characteristic are differ in number of jobs (i.e. 15, 25, 35, and 45 jobs), processing time variance, tardiness factor, relative range of due date. The results indicated that RS and SA are viable solution techniques that can yield good solutions to a range of scheduling problems, particularly, large-sized problems. B&B is a preferred technique in solving small-sized problems since

Table 2.1 Comparison of GA, SA, RS, and ACO

Problem	#Job	PTV	TF	DDR	B&B	GA % to B&B			SA % to B&B			RS % to B&B			RS Run time[&]	ACO % to B&B			ACO Run time[&&]
						Best	Median	Worse	Best	Median	Worse	Best	Median	Worse	(sec)	Best	Median	Worse	(sec)
Prob 401	15	L	L	N	90*	0.0	4.4	4.4	0.0	3.3	7.8	0.0	0.0	3.3	360	0.0	4.4	7.8	11.80
Prob 402	15	L	L	W	0*	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	360	0.0	0.0	0.0	0.35
Prob 403	15	L	M	N	3418*	0.0	0.0	0.0	0.0	0.0	0.8	0.0	0.0	0.2	360	0.0	1.1	2.1	13.5
Prob 404	15	L	M	W	1067*	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	12.7	360	0.0	0.0	0.0	11.05
Prob 405	15	H	L	N	0*	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	360	0.0	0.0	0.0	0.35
Prob 406	15	H	L	W	0*	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	360	0.0	0.0	0.0	0.30
Prob 407	15	H	M	N	1861*	0.0	0.0	0.0	0.0	0.0	0.4	0.0	0.0	0.4	360	0.0	0.0	1.1	13.75
Prob 408	15	H	M	W	5660*	0.0	0.0	0.9	0.0	0.0	0.6	0.0	0.0	0.9	360	0.0	1.1	1.5	13.20
Prob 501	25	L	L	N	264	0.0	1.5	3.8	0.8	1.9	4.2	0.8	0.8	1.5	960	-1.1	0.8	1.9	85.90
Prob 502	25	L	L	W	0*	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	960	0.0	0.0	0.0	1.70
Prob 503	25	L	M	N	3511	-0.4	0.2	0.9	-10.4	-9.9	-9.6	-0.4	-0.4	-0.4	960	-0.4	0.3	0.9	88.65
Prob 504	25	L	M	W	0*	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	960	0.0	0.0	0.0	1.50
Prob 505	25	H	L	N	0*	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	#	960	0.0	0.0	0.0	1.60
Prob 506	25	H	L	W	0*	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	960	0.0	0.0	0.0	1.60
Prob 507	25	H	M	N	7225	2.1	6.1	9.6	0.0	0.0	2.4	0.0	0.1	0.2	960	0.7	1.8	3.7	126.90
Prob 508	25	H	M	W	2067	-5.9	-5.9	-1.5	-7.4	-7.4	-3.1	-7.4	-7.4	-7.4	960	-5.9	5.9	14.2	102.90
Prob 601	35	L	L	N	30	76.7	150.0	193.3	20.0	43.3	96.7	20.0	31.7	46.7	1800	-46.7	-13.3	6.7	386.50
Prob 602	35	L	L	W	0*	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1800	0.0	0.0	0.0	3.75
Prob 603	35	L	M	N	17774	-0.7	0.4	2.2	0.1	0.8	1.4	0.1	0.2	0.7	1800	-0.5	0.1	0.3	776.65
Prob 604	35	L	M	W	19277	0.2	1.0	2.6	-0.2	0.7	2.8	-0.2	-0.1	0.1	1800	-0.8	0.3	1.3	382.05
Prob 605	35	H	L	N	291	13.7	37.3	56.7	-6.2	-1.2	8.9	-6.2	-4.1	-2.1	1800	-15.1	-8.8	-1.0	413.10
Prob 606	35	H	L	W	0*	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1800	0.0	0.0	0.0	3.65
Prob 607	35	H	M	N	13274	5.0	6.6	7.6	-1.7	-0.1	1.7	-1.7	-0.8	-0.2	1800	-1.4	-0.1	0.6	715.45
Prob 608	35	H	M	W	6704	-29.0	-28.6	-26.7	-29.4	-29.0	-26.9	-29.4	-29.2	-29.0	1800	-29.4	-24.8	-20.1	880.35
Prob 701	45	L	L	N	116	57.8	82.8	118.1	1.7	29.3	40.5	1.7	22.4	28.4	3600	-11.2	-5.6	0.0	1197.95
Prob 702	45	L	L	W	0*	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3600	0.0	0.0	0.0	11.05
Prob 703	45	L	M	N	27097	-0.1	1.5	2.5	-1.3	0.2	1.3	-1.3	-0.2	0.1	3600	-1.6	-1.0	-0.5	2638.25
Prob 704	45	L	M	W	15941	-2.4	-1.6	1.0	-3.3	-1.2	1.0	-3.3	-2.7	-1.8	3600	-2.8	-1.1	-0.3	1886.65
Prob 705	45	H	L	N	234	53.4	89.3	114.5	8.5	20.5	49.1	8.5	18.8	23.1	3600	-5.1	5.6	15.4	1168.85
Prob 706	45	H	L	W	0*	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3600	0.0	0.0	0.0	9.60
Prob 707	45	H	M	N	25070	-1.1	1.8	6.3	-3.4	-2.5	-0.8	-3.4	-2.9	-2.6	3600	-4.2	-3.3	-2.9	2524.65
Prob 708	45	H	M	W	24123	2.8	7.2	10.1	-4.0	-3.2	-2.0	-4.0	-3.9	-3.3	3600	-3.2	-1.7	-0.6	2336.10

* Indicates optimal solution
[&] Pentium 90MHz PC
[&&] Pentium 90MHz PC
[#] Divides by 0, the worst solution is 6.0
[@] Legend: PTV = Processing Time Variance, TF = Tardiness Factor, DDR = Due Date Range
L = Low, H = High, M = Moderate, N = Narrow, W = Wide

it is the only technique that can guarantee an optimal solution but require long running time.

Gagné *et al*. (2001) presented an *Ant Colony Optimization* (ACO) algorithm for solving single machine total tardiness with sequence-dependent setup time. They validate the performance the proposed algorithm by compare the results with those previously tested problems by Tan *et al*. (2000). The experimental result indicated that the performance of ACO and RS are competitive. It can be concluded that the RS performs better on small-sized problem while ACO is better for the larger problems. Gagné *et al*. (2001) also test the performance of ACO on larger test problem, i.e. 55, 65, 75, 85. The test problems sets and results are available on the website http://depcom.uqac.uquebec.ca/~3gagne/.

Table 2.1 presents the comparison of results of the four techniques, i.e. GA, SA, RS, ACO, including the computational time and the percentage differences between the median results of B&B algorithm and different heuristics.

The second focused topic in this paper is the application of a new meta-heuristic called *Particle Swarm Optimization* (PSO) to single machine total tardiness problem with sequence-dependent setup time. The Hybrid-PSO algorithm, by integrating standard PSO with Local Improvement Technique, is proposed. The objective of this study is to provide an efficient algorithm for this type of problem that yield the optimal or near-optimal solution with short computational time and can be applied to solve large problem size.

In order to summarize the previous research on single machine scheduling problems, Table 2.2 presents a list of related single machine scheduling problem with different objective functions that appeared in the literature.

Table 2.2 List of related single machine scheduling problem in the literature

Objective Function	Condition(s)	Setup time (ST_{ji})	Setup Cost (SC_{ji})	Earliness Cost(h_i)	Tardiness Cost(w_i)	Authors
SETUP COST PROBLEM						
Min no. of changeovers	tardiness is not allowed	0	1	0	M	Glassey (1968)
Min seq-dep changeover cost	tardiness is not allowed	0	Seq-dep	0	M	Driscoll and Emmons (1977)
Min sum of changeovers and inventory holding cost	idle time is allowed tardiness is not allowed	0	1	$h_i = h_j$ *for all i,j*	M	Gascon and Leachman (1988)
Min seq-dep changeover cost	$SC_{ji} = 1$ if $j<i$ $SC_{ji} = 0$ if $j \geq i$ tardiness is not allowed	0	$SC_{ji} = 1$ if $j<i$ $SC_{ji} = 0$ if $j \geq i$	0	M	Hu *et al*. (1987)
SETUP TIME PROBLEM						
Min seq-dep setup time	–	Seq-dep	0	0	0	Gavett (1965), Haynes *et al*. (1973), White &Wilson (1977)

Min number of setup time	–	1	0	0	0	Lockett and Muhlemann (1972)
Min total setup time	–	R	0	0	0	Liao and Yu (1996)
TOTAL TARDINESS AND SEQ-DEP SETUP TIME PROBLEM						
Min total tardiness with seq-dep setup time	–	Seq-dep	0	0	$w_i = w_j$ *for all i,j*	Rubin ans Ragatz (1995), Tan and Narasimhan (1997), Tan *et al*. (2000), Gagné *et al*. (2001)
EARLINESS/TARDINESS PROBLEM						
Min weighted tardiness	–	–	0	0	R	Potts and Wassenhove (1984) Tasgetiren *et al*. (2004)
Min weighted *e/t* penalties	–	0	0	R	R	Abdul-Razaq and Potts (1988), Ow and Morton (1989), Li (1997), Liaw (1999), Ibaraki and Nakamura (1994)
Min total *e/t* penalties	$h_i = ap_i$, $w_i = bp_i$	0	0	ap_i	bp_i	Yano and Kim (1991)
Min weighted *e/t* with seq-dep setup time	–	Seq-dep	0	R	R	Coleman (1992)
Min weighted *e/t* with commom due date	common due date	0	0	R	R	Mondal and Sen (2001), Azizoglu and Webster (1997)
Min total *e/t* with common due date and seq-dep setup time	common due date	Seq-dep	0	$h_i = h_j$ for all i,j	$w_i = w_j$ for all i,j	Rabadi *et al.*(2004)
Min total *e/t* penalties with inserted idle time	idle time is allowed	0	0	$h_i = h_j$ for all i,j	$w_i = w_j$ for all i,j	Chang (1999) , Ventura and Radhakrishnan (2003)
Min total e/t^2 penalties with inserted idle time	idle time is allowed	0	0	$h_i = h_j$ for all i,j	$w_i = w_j$ for all i,j	Shaller (2004)
Min weighted *e/t* penalties with inserted idle time	idle time is allowed	0	0	R	R	Chen and Lin (2002)
Min weighted *e/t* penalties with seq-dep setup cost and time	idle time is allowed	Seq-dep	Seq-dep	R	R	Sourd (2005)

R = Real data, a,b positive real number
M = Big positive number, p_i = processing time of job i

2.2 Vehicle Routing Problem with Time Windows (VRPTW)

VRPTW involves finding efficient routes for vehicles along the network to minimize or maximize pre-specified objective function. The set of routes for a fleet of vehicles, all starting and ending at a central depot, intended to serve a given set of customers, are determined. All customers may be visited only once by only one vehicle. Each customer must be served within a specified time interval or window. The lower and upper bounds of the time window define the earliest and latest time for beginning the service for the customer. Therefore, a vehicle is not allowed to begin service after the time window's upper bound. A waiting time is incurred if a vehicle reaches the customer before the time window's lower bound. Each customer has a specified service time. The total route time of a vehicle is the sum of its travel times (which are proportional to the distances traveled), waiting or idle times and the service times. The maximum route time should not exceed the maximum route time of each vehicle. The objective of VRPTW is to minimize the route length, the service cost, the travel time, and the number of vehicles or a combination of these depending upon the particular application.

The current VRPTW solution techniques can be categorized as exact algorithms, construction and improvement heuristics or meta-heuristics. An intensive survey of previous articles on VRPTW can be found in Desrochers *et al*. (1988) and Solomon and Desrosiers (1988). Recently, Bräysy *et al*. (2004) provided comprehensive surveys and compared previous VRPTW applications based on evolutionary algorithms.

'Exact' algorithms for VRPTW have been investigated by many researchers. Kolen *et al*. (1987) introduced the branch and bound method. Desrochers *et al*. (1992) used the column generation method to solve linear programming relaxation of the set partitioning formulation of the VRPTW. Fisher *et al*. (1997) presented an optimization algorithm based on K-tree relaxation and Lagrangian deposition methods.

Due to the massive computational requirement of exact algorithms, constructive heuristics have been introduced to 'build' the vehicle route. Solomon (1987) was the first one to introduce a variety of route construction heuristics. His results show that a sequential time-space based insertion algorithm outperforms other techniques. While Potvin and Rousseau (1993) reported that the use of parallel construction philosophy can substantially improve Solomon's results. Solomon's test problems have been used by several researchers as a standard benchmark problem for VRPTW. For the improvement heuristics, the algorithm based on edge exchange is suggested by Lin and Kernighan (1973). Potvin and Rousseau (1995) proposed a new 2-opt exchange heuristic. Savelsberg (1990) introduced the local search technique based on the *k*-exchange concept.

Meta-heuristics have been applied by many researchers. The search schemes of meta-heuristics are mainly based on simulating nature and on artificial intelligence. The strategies explore the search space more thoroughly in order to avoid local optima. Meta-heuristics include GAs (Potvin and Bengio, 1996), simulated annealing (Chiang and Russell, 1996), tabu search (Potvin *et al*., 1996), (Taillard *et al*., 1997) and parallel tabu search (Garcia *et al*., 1994). Some authors reported the hybridization of meta-heuristics. Thaniah *et al*. (1994) introduced GenSAT which is the hybrid combination of GAs, simulated annealing, tabu search and local search techniques. The Route-Neighborhood-Based Two-Stage metaheuristic (RNETS) was proposed by (Hwa *et al*., 1999). The concepts of nested parallel route construction and end handling are introduced. Gambardella *et al*. (1999) applied a multiple ant colony system in which the first colony minimizes the number of vehicles while the second colony minimizes the distance traveled. Tan *et al*. (2001) also developed and enhanced various meta-heuristics including

simulated annealing (with updated cooling scheme), a variant of tabu search ('strict' tabu) and a GA (with new crossover operations, hybrid hill-climbing and adaptive mutation). Homberger and Gehring (2005) presented a hybridization of two-phase meta-heuristics which combines (μ,λ) evolution strategy and tabu search heuristics.

2.3 Staff Planning and Scheduling and Home Care Research

2.3.1 Staff Planning and Scheduling Problem

The research on staff planning and scheduling has been studied extensively. They are commonly related to determination of the proper number of staff with particular skills and allocation of staff according to demand. The objectives are to minimize costs, meet customers' demands, satisfy employee preferences, or equally distribute the work shift, etc. Examples of staff planning and scheduling problem include airline crew scheduling (Yan and Tu, 2002, Goumopoulos and Housos, 2004), bus or truck driver scheduling (Bianco *et al*., 1992), nurse scheduling (Cheang *et al*., 2003), call center scheduling (Lin *et al*., 2000), etc. For further details of the staff planning and scheduling problems, refer to Ernst *et al*. (2004).

2.3.2 Home Care Research

Community care, home care, and also known as domiciliary care, is the social care service provided for the people in communities. The provision of home care service is increasing in UK and also in many countries around the world. The service is to provide care and support needed to assist people, i.e. older people, people with illness or physical disability, to be able to live as independently as possible in their own homes rather than use residential or nursing care. Community care includes very high tech kinds of care as well as simpler assistance. Care workers visits clients in their own home and help with daily tasks such as getting up, dressing, toileting, personal hygiene, provision of meals, housework, contact & befriending, shopping, and professional tasks, such as, medication, physical therapists, etc. Since the community care service provider is non-profitable business and due to the limited resources, the efficient management and many community care provision issues, i.e. outsourcing decision, strategic and operations management decision, arise in order to achieve high quality care service at low cost.

The research on the health care system in Sweden has been reported by Eveborn *et al*. (2006). The local authorities are facing increasing cost due to the rise in the number of older people and those who required support. A decision support system called *LAPS CARE* is developed to aid the staff planning task. The repeated matching approach was used as an optimization routine to obtain the solution for routing problem. They showed the considerably savings in planning time, quality of routes, and quality of service.

The third focused topic in this paper is related to the scheduling of care workers by considering the real case of Home care service in UK. The care worker scheduling problem is an extension of the Vehicle Routing Problem with Time Windows (VRPTW) with limited route time, even though some specific characteristics are different. VRPTW is a combinatorial optimization problem involving extremely large search spaces with correspondingly large numbers of potential solutions. Since the complexity class of the problem is NP-hard (non polynomial time), using the exact method is computationally inefficient. The third topic aims at the development of Particle Swarm Optimization (PSO) to efficiently solve this problem and also improve the existing solution technique.

The care worker scheduling problem in this paper was originally considered by *The Welsh Systems Consortium*; a partnership between seven local government authorities, the National Assembly for Wales, and a software supplier. The care worker schedule is a part of its attempt to develop and implement a fully integrated health and social care information system. The existing system, which is basically a manual approach, is used to develop a feasible and sound schedule. The Advanced Internet & Emergent Systems (AiMES) Centre at the University of Liverpool (www.AiMES.net) has been providing help to the Consortium and identified the potential improvement of the schedule of care workers. AiMES applied the proprietary software ILOG™ Dispatcher (http://www.ilog.com) and utilized its embedding features to develop the scheduling engine. AiMES has conducted an experiment with the different 'solver' methods within ILOG to obtain the best result.

The *savings heuristic* is used to construct the initial feasible route. The principle is the trade off between using more vehicles with shorter routes and fewer vehicles with longer routes. Figure 2.1(a) shows an example of two different ways of constructing visits *a* and *b*. The initial route can be further improved by using neighborhoods to reduce costs of the route. The following pre-defined neighborhoods are provided by ILOG Dispatcher: *2-opt*, *Or-opt*, *Relocate*, *Cross*, *Exchange*.

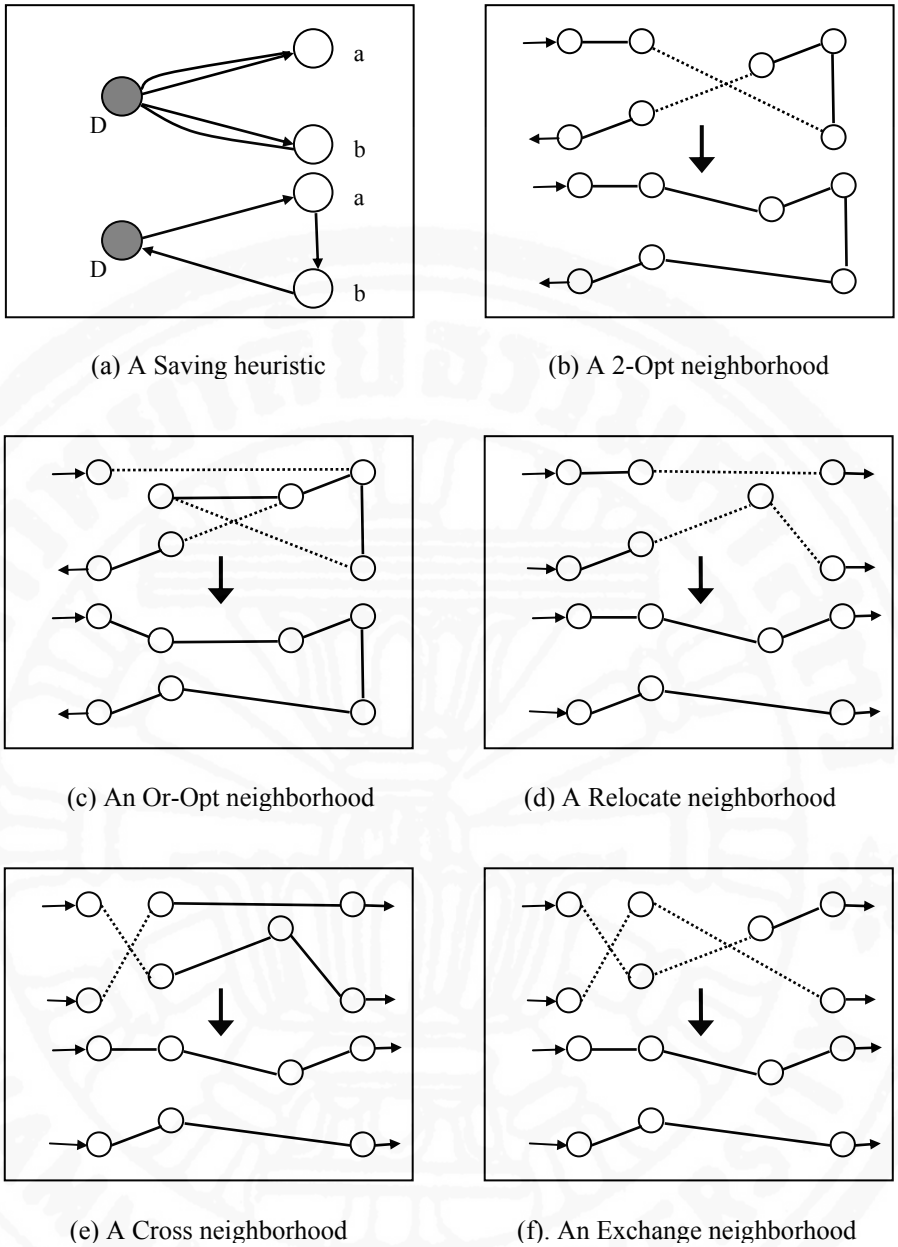
In *2-opt* neighborhood, two arcs are removed from the route and reconnected at any other possible parts of the route to improve the total route cost. The *Or-opt* neighborhood begins with move parts composed of one visit elsewhere in the route until all moves have been tested. Followed by, try moving parts composed of two consecutive visits until all moves have been tested. Then, try moving parts with three consecutive visits and so on. A *Relocate* neighborhood is the inter-route improvement in which a visit is inserted into another randomly selected route on condition that all constraints are satisfied. In a *Cross* neighborhood the ends of two routes are exchanged. The first part of one route is connected to the last part of another route and vice versa. The *Exchange* neighborhood swaps two visits of two routes provided that all constraints are not violated. Figure 2.1(b) to 2.1(f) shows the schematic view of all neighborhood procedures. The results from AiMES are much better than ones from Welsh Systems Consortium.

2.4 Scheduling Approaches

Two scheduling approaches are studied in this research. They are *Branch and Bound* (B&B) and *Particle Swarm Optimization* (PSO) algorithms.

2.4.1 Branch and Bound Algorithm

The Branch and Bound concept can be outlined with a simple term "an enumeration of a decision tree". A node represents a partial completed solution. A branch represents an alternative possible decision. The general Branch and Bound algorithm can be illustrated as follows.



(a) A Saving heuristic (b) A 2-Opt neighborhood

(c) An Or-Opt neighborhood (d) A Relocate neighborhood

(e) A Cross neighborhood (f). An Exchange neighborhood

Figure 2.1 Pre-defined initial solution Heuristic and neighborhood

Problem representation

A node represents the sub-problem. An arc represents the total cost of that node.

Initialization

The initial solution can be obtained from any heuristic techniques. For fast elimination of node, the objective value of the initial solution will be set as the upper bound for the problem.

Branching

Two branching schemes are normally used, depth-first branching or best-first branching rule. The node with the lowest objective value will be selected for branching. If there is more than one node, the depth-first branching will select the sub-problem at the greatest depth first. Follow the good objective value right to the bottom of the search tree. The best-first branching will branch the node with the same level first and then select the best one among all constructing nodes to search down to the bottom. This method tends to encounter good partial solutions all over the search tree. Then, eliminate as many newly created nodes as possible by fathoming method.

Bounding

For each newly created node, the objective value is calculated. The objective value is the sum of objective value of its parent node and the cost of that node. In this step, many heuristic techniques can be used to determine the lower bound, i.e., Lagrangian relaxation, Multiplier Adjustment Procedure, etc.

Fathoming

A node may be fathomed in three ways. First, a node is fathomed when it has been separated and all its successors have been generated. Second, a node is removed if its objective value is greater than or equal to the current upper bound for the problem. Third, a node is eliminated by other elimination techniques such as Lower Bound calculation, Dominance Criteria, or Adjacent Pairwise-Interchange, etc.

Stop

If there are nodes still remain unfathomed go to branching step. Otherwise, stop. The current node will be the optimal solution.

Many authors have been applied Branch and Bound algorithm to single machine scheduling problem since it is the technique that yield the global optimal solution. Li (1997) developed a Branch and Bound algorithm for the weighted earliness/tardiness problem. The algorithm is based on decomposition of the problem into two sub-problems. The lower bound of each sub-problem is obtained using Lagrangian relaxation with multiplier adjustment procedure. Liaw (1999) proposed a similar technique in which the lower bounds are computed using Lagrangian relaxation with multiplier adjustment and single pass procedure. The upper bound is obtained using two-phase heuristic procedures. Simple dominance rules are also derived to help eliminating nodes in the branch and bound search tree. Yano and Kim (1991) introduced dominance criteria to eliminate node in the branch and bound procedure. Recently, Sourd (2005) also used branch and bound procedure for the weighted earliness/tardiness problem with sequence-dependent setup time and cost between groups of products. Sourd (2005) used different form of problem representation so that the idle time can be inserted into the sequence. The time-indexed formulation with different relaxations was proposed to obtain the lower bound for the problem.

2.4.2 Particle Swarm Optimization Algorithm

Particle Swarm Optimization algorithm (PSO) is an evolutionary computation technique which originally developed by Eberhart and Kennedy (1995) and Kennedy and Eberhart (1995). PSO is motivated by the simulation of social behaviors such as bird flocking and fish schooling etc. Each individual, called particle represents the point in the search space. The population, called swarm, represents the set of points in which all points are the potential solution of the problem. The PSO concept is based on the sharing of information of its own previous experience and the group's experience. As a result, each particle adjusts the position toward its own previous best position (*pbest*) and to group's previous best position (*gbest*).

The principles of PSO algorithm can be stated as follows. PSO is initialized with a group of random particles (solutions) and velocities on n dimensions in the problem space. The d^{th} particle is represented as $X_d = \{x_{1d}, x_{2d}, ..., x_{nd}\}$. The performance of each particle is evaluated using a pre-defined fitness function, which depends on the problem to be solved. In every iteration, the position giving the best fitness value of d^{th} particle is recorded as previous best and represented as $pbest_d = \{pbest_{1d}, pbest_{2d}, ..., pbest_{nd}\}$. The best value obtained so far by the whole swarm is tracked and memorized as global best or *gbest*, where *gbest* represents the index of the best particle. PSO searches for optima by updating generations. The particle flies toward a new position by calculating a rate of position change or velocity which is represented by $V_d = \{v_{1d}, v_{2d}, ..., v_{nd}\}$. The historical information is used to calculate the velocity and the particle will change the position according to the following equation.

$$V_{nd} = \chi \left[w \times v_{nd} + c_1 \times r_1 \times (pbest_{nd} - x_{nd}) + c_2 \times r_2 \times (pbest_{n,gbest} - x_{nd}) \right] \quad (2.1)$$

$$x_{nd} = x_{nd} + v_{nd} \quad (2.2)$$

The velocity equation (Eq. 2.1) consists of three main parts. The first part is the momentum part. The velocity has to change from the current velocity. The second part is the cognitive part, which takes into account the personal experience of each particle. The third part is the social part which represents the inter-sharing of experience among particles. For the first part, the inertia weight (w) is employed to control the impact of the previous history of velocity on the current velocity. A large inertia weight facilitates global exploration while small value facilitates local exploration. Thus, a proper value of inertia weight provides balance between exploration and exploitation ability of the swarm. Shi and Eberhart (1998) analyzed the impact of inertia weight on the PSO performance. They suggest that the inertia weight should be initially set to a large value, and gradually decrease in order to fine-tune the search area. The w is usually decreased from 0.9 to 0.4 during a run. The variable r_1 and r_2 are random real number from uniform distribution $U(0,1)$. The acceleration constants c_1 and c_2 represent the weights of the stochastic acceleration terms that pull each particle toward *pbest* and *gbest* positions. A high value allows particle to make a rapid movement toward *pbest* and *gbest* positions while low value allows particle to move far from target region before coming back. For almost all applications, c_1 and c_2 are normally set to 2. Trelea (2003) recommend the constant inertia weight of 0.7968 and acceleration constants of 1.4962.

The intensive review and suggestion about the parameter selection of PSO can be found in Eberhart and Shi (2001). The application of constriction factor (χ) is suggest by Clerc (1999) to multiply with the whole terms in order to insure the convergence of PSO. The constriction factor can be defined as:

$$\chi = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|} \quad , where \ \varphi = c_1 + c_2, \ \varphi > 4 \tag{2.3}$$

When constriction factor is used, φ is normally set to 4.1, which make χ equal to 0.729. Moreover, the travel distance of particle is controlled by restricted the maximum velocity, v_{max}, in order to avoid the particle moving pass good solution. The v_{max} is served as a constraint that controls the maximum global exploration of PSO. If v_{max} is too high, the particle may fly passing good solution. If it is too low, the particle may not explore sufficiently which may result in trapping in local optima. After the particle move into new position, the performance of each particle is evaluated according to the pre-defined objective function. The same procedures are repeated until the maximum number of iteration is reached.

PSO has been successfully applied to many areas of applications; continuous function optimization (Eberhart and Kennedy, 1995, Kennedy and Eberhart, 1995), flowshop scheduling (Tasgetiren *et al*., in press), job shop scheduling (Lian *et al*., 2006), resource constrained project scheduling (Zhang *et al*., 2006), part-machine grouping (Andrés and Lozano, 2006), task allocation (Yin *et al*., in press), and neural network training (shen, 2006).

In addition, PSO has been applied successfully to various types of scheduling problem. This section also gives the insight review of the previous applications of PSO to various types of scheduling problems. The development of the algorithm, authors, successful applications, and the comparison with previous techniques, are summarized as follows:

A Single Machine Scheduling Problem is the first report of application of PSO to scheduling problem. Tasgetiren *et al*. (2004) applied PSO to the single machine total weighted tardiness problem. A heuristic called the smallest position value (SPV) rule was developed to translate continuous position value of PSO to discrete job sequence. A local search procedure, called variable neighborhood search (VNS) further improved the performance of PSO. The proposed algorithm was tested against ant colony optimization (ACO) and iterative local search (ILS) to give the best results reported in the literature. They concluded that PSO is as good as ACO and ILS.

For a Flowshop Scheduling Problem, Tasgetiren *et al*. (in press) used the same heuristic as single machine scheduling, i.e. SPV and VNS. They showed that 57 out of 90 and 195 out of 800 best known solutions were improved for total flowtime and makespan criterion respectively. Lian *et al*. (2006) embedded the new crossover operators in the original PSO and reported superior results over standard genetic algorithm (GA) for makespan objective. Liao *et al*.(in press) extended the discrete version of PSO to solve flowshop scheduling problem. The computational results showed the advantage over continuous based PSO by Tasgetiren *et al*. (in press) and GA. Another version of flowshop scheduling problem is studied by Allahverdi and Al-Anzi (2006). They applied PSO to the assembly flowshop scheduling problem. The objective of the study was to compare the performance of PSO with tabu search and the EST heuristic. The computational analysis indicates that tabu search outperforms others for the case when the due dates range is relatively wide. It also indicates that the PSO significantly outperforms others for difficult problems, i.e., tight due-dates. For the difficult problems, the inclusion of a dominance relation helps reduce the error by 65%.

For a Job Shop Scheduling Problem, Xia and Wu (2005) implemented the hybrid PSO and Simulated Annealing algorithm. The results have shown that the proposed

algorithm is a viable and effective approach for this type of problem, especially for large problem size.

For a Traveling Salesman Problem (sub-problem of vehicle routing problem), many versions of the PSO have been reported to generate satisfactory results. Pang *et al*. (2004a) used fuzzy matrices in representing the position and velocity of the particle. Another version of PSO for TSP was introduced also by Pang *et al*. (2004b). They implemented the space transformation to map Cartesian continuous space to the permutation space and applied local search and chaotic operations to improve the solution quality. Lopes and Coelho (2005) presented a hybrid model based on PSO and fast local search for the blind traveling salesman problem. The order crossover concept from GA was used to move the particles across the search space.

For a Flexible Manufacturing System Scheduling, Jerald *et al*. (2005) compared 4 approaches, i.e. a genetic algorithm (GA), simulated annealing, a memetic algorithm, and PSO for scheduling a flexible manufacturing system with multiple objective functions. They found PSO to be superior.

For a Project Scheduling Problem, Zhang *et al*. (2006) applied PSO to resource-constrained project scheduling to minimize total project duration. The performance of PSO is compared with the three heuristics (minimum total float, shortest activity duration, and minimum late finish time) and GA. The computation results indicated that PSO can obtain better results than the heuristic methods whilst achieving the same results as the GA, although PSO required fewer search iterations.

PSO has many desirable characteristics:

i. The concept of PSO is very simple and can be easily implemented to many applications.
ii. It is versatile, robust and general purpose in that it can be applied to similar versions of the problem with minor modification.
iii. It is computationally efficient in the sense that reasonable solutions can be obtained in short computational time.
iv. Unlike GA it involves only a few parameters so that it is easier to find the optimal combination of parameter values.
v. It is easily integrated with many local search techniques to improve the solution quality.
vi. The algorithm is well suited for parallelization by implementation on a cluster of workstations.

However, the only drawback of PSO is a fast convergence which can cause the solution to trap in local optima. This can be overcome by the application of the any Local Improvement Procedure (LIP), i.e. *Insertion and Swap*, or heuristic technique.

The successful application of PSO to scheduling and its favorable characteristics indicate that PSO is potentially suitable for scheduling problem.