

# An Improved Particle Swarm Optimization Algorithm for Care Worker Scheduling

**Chananes Akjiratikarl**

Industrial Engineering Program

Sirindhorn International Institute of Technology, Thammasat University, Pathum Thani 12121, THAILAND

Tel: +662- 986-9009 ext. 2107, Fax: +662-986-9112

**Pisal Yenradee<sup>†</sup>**

Industrial Engineering Program

Sirindhorn International Institute of Technology, Thammasat University, Pathum Thani 12121, THAILAND

Tel: +662- 986-9009 ext. 2107, Fax: +662-986-9112, E-mail: pisal@siit.tu.ac.th

**Paul R. Drake**

e-Business Division

The University of Liverpool Management School, Liverpool, L69 7ZH, UK

Tel: +44-151-795-3603, Fax: +44-151-795-3601, E-mail: drake@liverpool.ac.uk

Selected paper APIEMS2006

**Abstract.** Home care, known also as domiciliary care, is part of the community care service that is a responsibility of the local government authorities in the UK as well as many other countries around the world. The aim is to provide the care and support needed to assist people, particularly older people, people with physical or learning disabilities and people who need assistance due to illness to live as independently as possible in their own homes. It is performed primarily by care workers visiting clients' homes where they provide help with daily activities. This paper is concerned with the dispatching of care workers to clients in an efficient manner. The optimized routine for each care worker determines a schedule to achieve the minimum total cost (in terms of distance traveled) without violating the capacity and time window constraints. A collaborative population-based meta-heuristic called *Particle Swarm Optimization* (PSO) is applied to solve the problem. A particle is defined as a multi-dimensional point in space which represents the corresponding schedule for care workers and their clients. Each dimension of a particle represents a care activity and the corresponding, allocated care worker. The continuous position value of each dimension determines the care worker to be assigned and also the assignment priority. A *heuristic assignment* scheme is specially designed to transform the continuous position value to the discrete job schedule. This job schedule represents the potential feasible solution to the problem. The *Earliest Start Time Priority with Minimum Distance Assignment* (ESTPMDA) technique is developed for generating an initial solution which guides the search direction of the particle. *Local improvement procedures* (LIP), insertion and swap, are embedded in the PSO algorithm in order to further improve the quality of the solution. The proposed methodology is implemented, tested, and compared with existing solutions for some 'real' problem instances.

**Keywords:** Care Worker Scheduling, Meta-heuristic, Particle Swarm Optimization, Local Improvement Procedures, Heuristics, Home Care

## 1. INTRODUCTION

Home care (sometimes called domiciliary care) is part of the community care service that is the responsibility of local government authorities in the UK. Its aim is to provide the care and support needed to assist people, par-

ticularly older people, people with physical or learning disabilities and people who need assistance due to illness to live as independently as possible in their own homes. Home care is a viable alternative to in-hospital, residential or institutional based nursing care and the underlying driver is that it leads to a higher quality of life for the cli-

---

<sup>†</sup> : Corresponding Author

ents as well as lower costs for the client and the state. Care workers visit clients in their own homes and help them with not only the simpler daily tasks such as getting up, dressing, toileting, personal hygiene, provision of meals, housework, shopping, contact and befriending but also tasks such as medication and physical therapy that require higher levels of skills.

Until recently, homecare in the UK has been performed predominantly by care workers employed by the social services departments of local government authorities i.e., in-house. However, local authorities are increasingly outsourcing homecare from the independent sector, with some outsourcing as much as 100%. This is being driven in the first instance by central government seeking to achieve *best value* where value can be seen as the ratio of quality to cost.

Care workers travel from their own homes to deliver care to their allocated clients at a specified time or within a specified time window, and they return home after finishing their visits. The scope of this article deals with the dispatching of care workers to clients in an efficient manner under time and capacity constraints. Optimization techniques are developed to schedule the care workers on a daily basis to minimize the distance traveled.

The benefits of efficient scheduling of care workers are:

- i: Reduce the traveling distance and hence traveling costs of the care workers.
- ii: Improve worker utilization by reducing the 'waste' of travel and consequently reduce the number of workers required.
- iii: Improve working conditions for care workers by improving the shift pattern.
- iv: Increase customer service by satisfying all service requirements within specified time windows.
- v: Combined with automation through computerisation, a scheduler would free-up care managers to undertake a more regulatory role and to focus on more strategic issues.

The care worker scheduling problem can be modeled by the *vehicle routing problem with time windows* (VRPTW) with limited route time, even though some specific characteristics are different. In general, the vehicle routing problem involves finding efficient routes for vehicles within a network to minimize or maximize a pre-specified objective function. VRPTW is a combinatorial optimization problem involving extremely large search spaces with correspondingly large numbers of potential solutions. The complexity class of the problem is *NP-hard* (non-polynomial time) in which finding optimal solutions is difficult. A collaborative population-based meta-heuristic called *Particle Swarm Optimization* (PSO) is proposed and specifically designed to solve the problem. An initial solution heuristic and local search techniques are embedded in the PSO algorithm in order to achieve better solution quality.

To the authors' knowledge, this is the first time that

PSO has been applied to care worker scheduling. Nevertheless, some heuristic techniques are known to apply to similar types of problems.

The remaining sections of this article are organized as follows: The next section intends to give some background to PSO and VRPTW as well as a review of the application of PSO and previous solution techniques for related problems. Section 3 explains descriptions of the problem including the objective function and constraints. Section 4 illustrates the methodological approach of application of PSO to care worker scheduling, followed by the computational experiments in Section 5. Finally, the conclusions and a direction for future research are presented.

## 2. LITERATURE REVIEW

### 2.1 Application of PSO to Scheduling

The Particle Swarm Optimization (PSO) algorithm is an evolutionary computational algorithm originally developed by Eberhart and Kennedy (1995) and Kennedy and Eberhart (1995). It is a population-based searching technique that simulates the social behavior of birds flocking or fish schooling, etc. Each individual, called a *particle* represents a point in the search space. The population, called a *swarm*, represents the set of points that are potential solutions. PSO is based on the interaction and the social communication of the group of particles. The particle iteratively flies through the search space by using the velocity function, which is constantly updated based on its own previous experience and the group's experience. Each particle tends to adjust the position toward its own previous best position and to the group's previous best position. Tracking and memorizing the best positions encountered builds the particle's experience. PSO possesses a memory i.e. every particle remembers the best position it has reached. PSO combines local search (through self experience) with global search (through neighboring experience), attempting to balance exploration and exploitation.

PSO has been applied successfully to various types of scheduling problem. Tasgetiren *et al.* (2004) were the first to report the application of PSO to scheduling. They used PSO to solve the single machine total weighted tardiness problem. A heuristic called the *smallest position value* (SPV) rule was developed to enable PSO to be applied to scheduling. 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. The results have shown that three approaches are able to find the optimal and best known solutions for all problem instances in reasonable CPU time. They concluded that PSO is as good as ACO and ILS. The use of

VNS significantly improves the solution quality for large problems.

Tasgetiren *et al.* (2006) applied PSO to permutation flowshop sequencing also. The objective was to minimize makespan and total flowtime. They used the same heuristics as before, i.e. SPV and VNS. The algorithm was evaluated against benchmark. For the total flow time criterion, a total of 57 out of 90 best known solutions were improved while for makespan, a total of 195 out of 800 best known solutions were improved.

The application of PSO to the assembly flowshop scheduling problem has been addressed by Allahverdi and Al-Anzi (2006). The objective of the study is to compare the performance of PSO with tabu search and the EST heuristic. The computational analysis indicates that tabu search outperforms the others for the case when the due dates range is relatively wide. It also indicates that the PSO significantly outperforms the 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%.

Xia and Wu (2005) presented a hybrid PSO approach for the multi-objective flexible job-shop scheduling problem. The *hybridizing of PSO and Simulated Annealing* algorithm is implemented. The results obtained from the computational study have shown that the proposed algorithm is a viable and effective approach for this type of problem, especially for large problem size.

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.

Zhang *et al.* (2006) applied PSO to resource-constrained project scheduling. The objective is 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 a 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. It is very simple and can be adapted to many areas of application. It is versatile, robust and general purpose since it can be adapted quite simply with minor modification. Unlike a GA it involves only a few parameters so that it is easier to find the best combination of parameter values. Furthermore, it is computationally efficient and as a population-based heuristic it is amenable to parallel implementation.

The successful application of PSO to scheduling and its favourable characteristics indicate that PSO is potentially suitable for care worker scheduling.

## 2.2 Previous Solutions for VRPTW

VRPTW involves the determination of an efficient set of routes, all starting and ending at a central depot, for

a fleet of vehicles intended to service a given set of customers. All customers may be visited only once by only one vehicle. Each customer must be serviced 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, 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 author 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 Rusell, 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  $(\mu, \lambda)$  evolution strategy and tabu search heuristics.

The number of possible solutions for VRPTW grows exponentially with the problem size. Since the complexity of VRPTW is NP-hard, the improved PSO algorithm is proposed to efficiently solve the problem.

### 2.3 Existing Techniques for Care Worker Scheduling Problem

The care worker scheduling problem was first considered by *The Welsh Systems Consortium*, a partnership between seven local government authorities in Wales. The traditional method, i.e. manual calculation, is used to obtain the solution. Later on the Advanced Internet & Emerging Systems Institute (<http://www.aim.es.net>) at the University of Liverpool attempted to develop the care worker scheduling engine in order to improve the existing solutions obtained by the Welsh Systems Consortium. AiMES applied the proprietary software ILOG<sup>TM</sup> Dispatcher 4.0 (<http://www.ilog.com>) and utilized its embedding features to develop the scheduling engine. ILOG<sup>TM</sup> provides a variety of pre-defined first solution heuristics for use in generating an initial solution and also offers a variety of pre-defined neighborhoods for use in improving a solution. 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. The initial route is further improved by using pre-defined neighborhoods to reduce the costs of the route, these are *2-opt*, *Or-opt*, *Relocate*, *Cross*, *Exchange*. The real care service data is used as a benchmark problems. The results obtained by AiMES outperform those obtained by the Welsh Systems Consortium in every case of the test problem instances. It produces the considerably saving in distance traveled.

## 3. PROBLEM DESCRIPTION

The aim is to schedule care workers on a daily basis, minimising the total distance traveled whilst satisfying all

constraints. The problem is defined as follows:

- i: Care workers are scheduled according to the requirements of the clients, who may require more than one activity/visit per day.
- ii: Demand has substantial peaks during some periods of the day i.e. morning and early evening.
- iii: Each activity must be delivered within a specified time window and location.
- iv: Each activity can involve only one visit by one care worker i.e. no activity splitting is allowed.
- v: Care workers start from their homes and return after finishing all assigned activities. The total traveling distance is the sum of the distance from care worker's home to the first client, the distances between the successive clients and the distance from the last client back to the worker's home.
- vi: For critical, medical activities the time window is a target time  $\pm 5$  minutes. For non-critical activities the window is  $\pm 15$  minutes.
- vii: The maximum capacity of each worker is 7.5 hours per day, including travel time.
- viii: In the model used here, each worker is assumed to be available 24 hours per day, but can be used for only 7.5 hours in that period.
- ix: The travel speed of a care worker is assumed to be 30 miles/hour and ignores traffic conditions.
- x: Locations are represented by easting/northing coordinates for each postcode. The Euclidean or straight-line distance is assumed between locations.
- xi: Some issues have been neglected in this first study, such as client-carer familiarity, skill-matching requirements, care plan shift patterns and male/female preference for care workers.

As mentioned earlier, care worker scheduling is a version of VRPTW but with some characteristics of its own. To be more specific, the problem can be seen as a special case of the Multi Depot Vehicle Routing Problem with Time Windows (MDVRPTW) in which each vehicle (a care worker) stationed at each corresponding depot (care worker's home). In particular, care worker scheduling is different to VRPTW in that all care workers (vehicles) start and finish at their own, separate homes, whereas for general VRPTW all vehicles start and end at the same central depot. For a discussion and a solution procedure of MDVRPTW, refer to Polacek *et al.* (2004).

## 4. METHODOLOGY

### 4.1 Particle Swarm Optimization (PSO)

PSO is a collaborative population-based search that models the social behaviour of particles. PSO is initialized with a group of random particles (solutions) and velocities in  $n$ -dimensional space. The performance of each

particle is evaluated using a pre-defined fitness function, which depends on the problem to be solved. At every iteration, the position giving the best fitness value of every particle is recorded as the previous best. The best value obtained so far by the whole swarm is memorized as the global best. PSO searches for optima by updating generations. The particle flies toward a new position by calculating a rate of position change or velocity. The historical information is used to calculate the velocity and the particle updates the position according to the following equations.

$$v_i^{t+1} = \chi [w \times v_i^t + c_1 \times r_1 \times (pbest_i - x_i^t) + c_2 \times r_2 \times (gbest - x_i^t)] \quad (1)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (2)$$

For the velocity equation (Eq. 1), the inertia weight ( $w$ ) is employed to control the impact of the previous history of velocity on the current velocity. The inertia weight is used to provide a balance between the exploration and exploitation ability of the swarm. Shi and Eberhart (1998) suggest that the inertia weight should be set initially to a large value to facilitate global exploration, and gradually decreased in order to fine-tune the search area. The  $w$  is usually decreased from 0.9 to 0.4 during a run. The variables  $r_1$  and  $r_2$  are random real numbers from the 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 particles to make a rapid movement toward  $pbest$  and  $gbest$  positions while a low value allows particles to move far from the target region before coming back. For almost all applications,  $c_1$  and  $c_2$  are normally set to 2. An intensive review and suggestions about the parameter values for PSO can be found in Eberhart and Shi (2001). The application of a constriction factor ( $\chi$ ) is suggested by Clerc (1999), to multiply with the whole terms in order to insure the convergence of PSO. The constriction factor is usually set equal to 0.729. Moreover, the travel distance of a particle is controlled by restricting the maximum velocity,  $v_{max}$ , in order to control the maximum global exploration of PSO. After the particles have moved into new positions, the performance of each particle is evaluated according to the pre-defined objective function. These procedures are repeated until the maximum number of iterations is reached.

#### 4.2 Earliest Start Time Priority with Minimum Distance Assignment (ESTPMDA)

An initial solution for the problem is determined by using ESTPMDA heuristics. Let  $m$  and  $n$  be the number of care workers and care activities, respectively. In the first stage,  $m$  care workers are sorted so that the route connecting the care workers' homes has the shortest distance. The purpose of sorting the care workers is to guide

the search of the particles. Then  $n$  care activities are arranged according to the earliest start time (EST) rule, which is in ascending order of their ideal start time. The priority value of assignment is represented by an  $(m \times n)$  matrix. The header of columns represents the EST activity list and the header of rows represents the care worker ID (after sorting). In the second stage, the procedure tries to insert activity, individually into all care worker routes and calculates the cost of insertion (additional distance) of each route. The cost of insertion is then translated to priority value, where the lower the cost of insertion means the higher the priority value. An example priority-matrix is shown in Table 1.

From Table 1, suppose there are 6 activities and 3 care workers. The activity will be assigned to the highest possible priority value route providing that the time windows and capacity constraints are not violated. From the matrix, activity 3 (J3) has the highest priority to assign to care worker 2. Similarly, activity 5 prefers to assign to care worker 3. If the infeasibility assignment occurred, activity will be assigned to the next highest priority care worker. Once the assignment is accomplished for each activity, the priority value is revised according to the real assignment and is kept in the matrix table for future reference. The procedure will repeat until all activities are assigned to care worker routes. It should be noted that the calculation of the priority matrix, assignment step, and the feasibility checking is done on a one-by-one basis.

**Table 1.** Priority matrix for each particle.

EST Priority	J3	J5	J4	J1	J2	J6
1 <sup>st</sup>	C2	C3	C2	C1	C1	C3
2 <sup>nd</sup>	C1	C2	C1	C3	C2	C1
3 <sup>rd</sup>	C3	C1	C3	C2	C3	C2

The above heuristic can be used to obtain one feasible solution. Since PSO is a population-based search technique, variants of feasible solutions must be obtained for each of the particles in the population. To do so, pairwise interchange (PI) is applied to the EST activity list. The parameter  $pchange$  (the percentage of selecting activities to do PI) controls the number of times PI is performed. It controls the amount of disturbance of the original EST sequence, which is set equal to 2% of the number of activities. The disturbance in EST sequence will result in a different assignment sequence. When the assignment sequence is changed, the order of activity to be assigned to care worker will change. The care worker route will be occupied differently by activities. Applying the same ESTPMDA procedure to this new activity sequence, a different initial feasible solution is obtained for each particle.

### 4.3 Application of PSO to care worker scheduling

In order to apply PSO to care worker scheduling, each particle represents a multi-dimensional point in space. The main variables in PSO can be summarized as follows:  $x'_{ijk}$  denotes the position value of dimension  $i^{th}$  and  $j^{th}$  of particle  $k^{th}$  in the swarm at iteration  $t$ . The  $i^{th}$  dimension ( $i = 1, 2, \dots, n$ ) refers to the EST list of activities, where  $n$  is the total number of the activities. The  $j^{th}$  dimension ( $j = 1, 2, \dots, m$ ) represents the care worker ID, where  $m$  is the total number of care workers. For each particle at iteration  $t$ ,  $x'_k = [x'_{11k}, x'_{12k}, x'_{13k}, \dots, x'_{n1k}, x'_{n2k}, \dots, x'_{nmk}]$ . The position value represents the priority value of the care worker to be assigned by the activity. The whole swarm is the set of all particles,  $x' = [x'_1, x'_2, x'_3, \dots, x'_k]$ , whereas  $k$  is the population size. The particle moves through the search space by using its associated velocity value,  $v'_{ijk}$ . The  $v'_{ijk}$  refers to velocity of dimension  $i^{th}$  and  $j^{th}$  of particle  $k^{th}$  at iteration  $t$ . The previous best position,  $pbest'_{ijk}$ , is the position value of the  $k^{th}$  particle with the minimum objective function so far at iteration  $t$ . The minimum objective value is recorded as  $pbest'_k$ . The best value obtained by the whole swarm is tracked and memorized as global best,  $gbest'$ , where  $gbest'$  represents the index of the best particle.

The PSO algorithm and its Pseudo code for care worker scheduling are presented below.

*Pseudo code: Improved PSO algorithm*

**Start**

**Apply Initial Solution Heuristic:** ESTPMDA (see section 4.2)

**Initialize PSO parameters:** Random  $x^0_{ijk}$  and translate to  $x^0_k$  matrix according to ESTPMDA heuristic,  $popsiz$ ,  $maxiter$ ,  $\chi$ ,  $w^0$ ,  $c_1$ ,  $c_2$ ,  $v_{max}$ ,  $v^0_k$ ,  $pbest^0_k$ ,  $gbest^0$  for all  $k$

**Initialize LIP parameters:**  $Nselect$ ,  $numinsert$ ,  $probac$

**Do {**

**For {  $k = 1$  to  $popsiz$**

**Solution representation:** using Heuristic assignment (see step2 of section 4.3.1); and Repair

**Fitness value evaluation:** calculate  $f(Assign(x'_k))$

**Apply Local Improvement Procedures:**

a. **Swap Procedure** (see section 4.4.1); and Repair

b. **Insertion Procedure** (see section 4.4.2); and Repair

**Update objective value after LIP**

**Update  $pbest$  and  $gbest$**

**Calculate velocity:** using Eq. 5

**Update position value:** using Eq. 6

**Update inertia weight:** using Eq. 7

**}End for**

**} while (termination)**

**End**

### 4.3.1 PSO Algorithm

#### Step 1. Initialization

The population of particles is initialized randomly. The number of particles is equal to the parameter  $popsiz$ , which is 10. A preliminary experiment was conducted to test the population sizes of 10 and 20 and found that the size of 10 is adequate. The continuous position values of each dimension of each particle are generated by using the following formula:

$$x^0_{ijk} = x_{min} + (x_{max} - x_{min}) \times U(0, 1) \quad (3)$$

where,  $x_{min}$  and  $x_{max}$  are the pre-defined range of position values, which are set to 0 and  $n$ , respectively. Note that  $n$  is the number of care activities.  $U(0,1)$  is a uniform random number in the range 0 to 1. The previous best position of each particle  $pbest^0_{ijk}$  is initially set equal to  $x^0_{ijk}$ . Similarly, their associated random velocities can be generated according to the following formula.

$$v^0_{ijk} = v_{min} + (v_{max} - v_{min}) \times U(0, 1) \quad (4)$$

where,  $v_{min}$  and  $v_{max}$  are the pre-defined range of the velocity values, which are set to  $-n/2$  and  $n/2$ , respectively.

Note that based on some experiments not presented in this paper, the values of  $x_{min}$ ,  $x_{max}$ ,  $v_{min}$ , and  $v_{max}$  do not

**Table 2.** Translation of the initial solution to PSO position value matrix.

Random position value for each particle						
EST \ Random value	J3	J5	J4	J1	J2	J6
1	1.2	4.2	5.2	2.8	1.4	1.3
2	2.5	2.1	1.5	4.2	0.2	5.1
3	3.9	3.4	0.5	3.6	3.7	2.4
Priority matrix for each particle						
EST \ Priority	J3	J5	J4	J1	J2	J6
1 <sup>st</sup>	C2	C3	C2	C1	C1	C3
2 <sup>nd</sup>	C1	C2	C1	C3	C2	C1
3 <sup>rd</sup>	C3	C1	C3	C2	C3	C2
New position value matrix for each particle ( $x^0_k$ )						
Carer ID \ EST	J3	J5	J4	J1	J2	J6
C1	2.5	4.2	1.5	2.8	0.2	2.4
C2	1.2	3.4	0.5	4.2	1.4	5.1
C3	3.9	2.1	5.2	3.6	3.7	1.3

significantly affect the performance of the algorithm.

### Translation of initial solution to PSO position value matrix.

The initial feasible solution obtained by the ESTPMDA heuristic is converted to the continuous position values. The random position values obtained from (Eq.3) are ranked in ascending order. The lower value refers to the higher priority. Recall the priority matrix from Table 1. Then, create a new position value matrix ( $x_k^0$ ) by mapping the position value according to the priority matrix, i.e. lowest value assigned to the highest priority. As a result, the initial position values that comply with the priority value matrix of all particles are obtained. An example of the translation procedure is shown in Table 2.

### Step 2. Heuristic Assignment Scheme (Solution Representation)

The heuristic assignment scheme is designed especially to translate the continuous position value into the discrete schedule. Each dimension of a particle represents a corresponding care activity and care worker ID. The position value implies the assignment priority of the activity to the care worker. In order to generate the schedule, the position values of each particle are sorted into ascending order to create the priority matrix. For example, in column J3 of Table 3, the position value 2.1 is the lowest followed by 3.4 and 4.2. This implies that activity J3 is preferable to care worker 2 (C2) as the first priority followed by care worker 1 (C1) and care worker 3 (C3) as the second and third priority, respectively. Repeat the same procedure for all care activities, the priority matrix will be created.

Then, the assignment of care activities to care worker route is performed. The assignment of care activity to care worker is performed on a one-by-one basis simultaneously with the feasibility checking. In order to do so, care activity is assigned to the care worker that has the highest priority provided that the time windows and capacity constraints are not violated. During assignment, the current activity and previously assigned activities are allowed to move themselves within time windows in order to make the schedule feasible. If infeasibility occurs, the candidate activity is assigned to the care worker with the next highest assignment priority. After all activities have been assigned, the position value matrix is revised to be compatible with the new assignment.

Table 3 illustrates the solution representation technique. For the Table, J3 and J5 are assigned to C2 and C1, respectively. J4 is preferable to assign to C2. After performing the feasibility checking, suppose infeasibility occurs due to time conflict with previously assigned job (J3). J4 has to be assigned to the next highest priority, which is C3. Then, the position value matrix of C2 and C3 has to be revised according to the real assignment as highlighted in Table 3. The procedure will be repeated until all care activities are assigned.

### Step 3. Fitness value evaluation

After the assignment of care activities to all care workers is accomplished, the fitness of each particle is evaluated. Let  $Assign(x_k^t)$  be the corresponding sequence at iteration  $t$  of particle  $x_k^t$ . The objective function value  $f(Assign(x_k^t))$  is calculated. The objective function is the sum of all distances traveled.

**Table 3.** Solution representation.

Position value matrix for each particle at iteration $t$ ( $x_k^t$ )						
EST \ Carer ID	J3	J5	J4	J1	J2	J6
C1	3.4	0.5	5.4	3.5	2.3	2.3
C2	2.1	3.5	1.6	4.3	3.6	1.2
C3	4.2	2.3	3.2	1.2	3.4	4.5
Assignment priority matrix						
EST \ Priority	J3	J5	J4	J1	J2	J6
1 <sup>st</sup>	C2	C1	C2	C3	C1	C2
2 <sup>nd</sup>	C1	C3	C3	C1	C3	C1
3 <sup>rd</sup>	C3	C2	C1	C2	C2	C3
Actual assignment (after feasibility checking)						
EST List	J3	J5	J4	J1	J2	J6
Carer ID	C2	C1	C3	C3	C1	C2
Position value matrix for each particle at iteration $t$ (after repairing)						
EST \ Carer ID	J3	J5	J4	J1	J2	J6
C1	3.4	0.5	5.4	3.5	2.3	2.3
C2	2.1	3.5	3.2	4.3	3.6	1.2
C3	4.2	2.3	1.6	1.2	3.4	4.5

### Step 4. Update previous best position value and previous best fitness value.

Initially, the previous best position value and previous best fitness value of particle  $k^{th}$ , i.e.  $pbest_{ijk}^0$  and  $pbest_k^0$  are set equal to the initial position value and the initial objective value of each particle, respectively. At each iteration  $t$ , the current fitness value  $f(Assign(x_k^t))$  is compared with the previous best fitness value  $f(Assign(pbest_k^{t-1}))$ . If  $f(Assign(x_k^t))$  is less than  $f(Assign(pbest_k^{t-1}))$ , set  $pbest_k^t$  equal to  $f(Assign(x_k^t))$  and  $pbest_{ijk}^t$  equal to  $x_{ijk}^t$ , which is the current position values of particle  $k^{th}$ .

### Step 5. Update global best's dimension index

The initial global best particle, i.e.  $gbest^0$ , refers to

the array index of particle which has the minimum value among  $pbest^0_k$ . Initially, the global best position value and the global best fitness value are equal to  $pbest^0_{ij,gbest}$  and  $pbest^0_{gbest}$ , respectively. At each iteration  $t$ , the  $gbest^t$  are updated. By comparing a new previous best fitness value with the global best fitness value, if  $f(Assign(pbest^t_k))$  is less than  $f(Assign(pbest^{t-1}_{gbest_{t-1}}))$ , set  $gbest^t = k$ .

#### Step 6. Velocity calculation

The velocity of each particle can be calculated by using the following formula.

$$v^{t+1}_{ijk} = \chi [w^t v^t_{ijk} + c_1 r_1 (pbest^t_{ijk} - x^t_{ijk}) + c_2 r_2 (pbest^t_{ij,gbest^t} - x^t_{ijk})] \quad (5)$$

The velocity is limited by a predefined maximum velocity ( $v_{max}$ ) which is set equal to  $n/2$ . If  $v^{t+1}_{ijk} > v_{max}$ , set  $v^{t+1}_{ijk} = v_{max}$ . Note that the maximum velocity is limited by  $v_{max}$  to prevent the position values of the particle from changing too rapidly. The constriction factor, inertia weight, and acceleration constant are set according to the recommended values of 0.729, 0.9, and 2.0, respectively.

#### Step 7. Update position value

The position value of each dimension of a particle is updated according to the Eq.6. Consequently, the particle moves toward a new position and the new solution is obtained. Note that the position values are updated without any restriction of  $x_{min}$  and  $x_{max}$ , which is used for initialization only.

$$x^{t+1}_{ijk} = x^t_{ijk} + v^{t+1}_{ijk} \quad (6)$$

#### Step 8. Update inertia weight

The inertia weight is updated at each iteration according to the following formula:

$$w^t = w^t \times \alpha \quad (7)$$

where, the decrement factor  $\alpha$  is 0.975.

#### Step 9. Termination

Repeat Steps 2 to 8 until the stopping criterion is met. For this application, the maximum number of iterations ( $maxiter = 20$ ) is the stopping criterion.

### 4.4 Local Improvement Procedures (LIP)

In order to overcome the fast convergence of standard PSO, the local improvement procedure (LIP) is applied during the searching phase. LIP is a simple and effective search algorithm, which can be applied easily to standard PSO. Since solutions encountered among particles during the search phase are differences, applying LIP encourages each particle to fine-tuned the search around its own searching area. It allows the particles to explore

the search space more thoroughly, increases the possibility of finding better solution, escapes from local minima, and improves the solution quality. After the fitness value evaluation step, LIP is applied to the global best solution and some randomly selected previous best solutions according to the specified number of selection ( $Nselect$ ), which is set equal to 35% of  $popsiz$ . The swap and insertion procedures are applied sequentially to the selected solutions.

#### 4.4.1 Swap Procedure

The swap procedure interchanges activities between care workers' routes. Given a set of care workers' routes  $S = \{R_1, R_2, \dots, R_p, \dots, R_q, \dots, R_m\}$ , each route consists of the sequence of service activities on that route. The procedure is done by selecting a pair of routes  $R_p$  and  $R_q$ . The routes are selected sequentially starting from  $R_1$  and  $R_2$ , and continue until  $R_{m-1}$  and  $R_m$ . Suppose the set of care activities in route  $R_1$  and  $R_2$  are represented by  $R_1 = \{a_{11}, a_{21}, \dots, a_{nR1,1}\}$  and  $R_2 = \{a_{12}, a_{22}, \dots, a_{nR2,2}\}$ , respectively. The notation  $a_{xy}$  refers to the care activities in position  $x$  of route  $y$ . The swap procedure of care activities is performed sequentially along these two routes. Beginning with exchanging the activities  $a_{11}$  and  $a_{12}$ , then  $a_{11}$  and  $a_{22}$ , and so on until  $a_{11}$  and  $a_{nR2,2}$ , followed by exchanging the activities  $a_{21}$  and  $a_{12}$ , then  $a_{21}$  and  $a_{22}$ , and repeating the swap procedure until  $a_{nR1,1}$  and  $a_{nR2,2}$ . In other words, exchange all pairs of activities sequentially from  $a_{x1}$ , where  $x = 1, 2, \dots$ , to  $nR1$ , with  $a_{x2}$ , where  $x = 1, 2, \dots$ , to  $nR2$ . In the swap procedure, the selected activity in one route may or may not be placed at the same position as the selected activity in the other route but it is placed at a position that is consistent with its time window. Only feasible moves, which do not violate capacity and time windows constraints, are accepted. After interchanging a pair of activities, the objective value is computed. The first-best strategy is used here. This accepts the first improving moves and adopts them as the new starting routes. In the case that moving results in the same objective value, the new route is accepted with a pre-defined probability. The acceptance probability ( $probaccept$ ) is set equal to 0.5. Once the interchange of all pair activities between  $R_1$  and  $R_2$  is accomplished, the next pair of routes  $R_1$  and  $R_3$  is selected. Swap all pairs of activities between route  $R_1$  and  $R_3$  in the same manner and continue with the next pair of routes. Continue the same procedure until all pairs of routes are selected, or in other words, select  $R_p$  and  $R_q$ , where  $p = 1, 2, \dots, m-1$  and  $q = p+1, p+2, \dots, m$ .

Once the swap procedure is accomplished, the position value matrix of all care activities will be revised according to new schedule.

#### 4.4.2 Insertion Procedure

A unique and efficient insertion procedure, which utilizes the priority matrix, is introduced. Recall the priority matrix from Table 3 section 4.3.1, the priority matrix represents the degree of attractiveness of each care worker's route to the activity. Following the priority ma-



**Table 4.** Test Results.

Test	Problem size	Welsh Consortium	AiMES	Improved PSO <sub>mean</sub> (Mean result)	Improved PSO <sub>best</sub> (Best result*)	%Saving (PSO <sub>best</sub> compare to Welsh Consortium)	%Saving (PSO <sub>best</sub> compare to AiMES)
1	106 Activities 12 Carers	591.5	348.7	328.3	306.8	92.8%	13.7%
2	101 Activities 12 Carers	642.7	384.4	353.3	332.3	93.4%	15.7%
3	106 Activities 12 Carers	658.0	416.8	378.0	351.8	87.0%	18.5%
4	111 Activities 12 Carers	1388.3	535.2	404.6	370.5	274.7%	44.5%
5	108 Activities 12 Carers	667.3	388.2	380.9	348.6	91.4%	11.4%

(Note) \* Best results over 20 replications.

trix, the care activities are removed one-by-one from the current route and inserted into the new route that has the next highest priority. Similarly, the selected activity is inserted into the new route at the position that is compatible with its time windows. Then, feasibility checking and fitness value evaluation are performed.

The procedure accepts the first improving move or the equivalent move with the probability *probaccept*, which is set equal to 0.5. If a better move is not found, the selected activity seeks the next highest priority route. The insertion procedure will not try to insert into every care worker's route since this would require a high computational effort. The number of care workers to be inserted is controlled by the parameter *numinsert*, which is set to 50% of the total number of care workers' routes.

After the insertion procedure is completed, the position value matrix is modified to be compatible with the new sequence.

The insertion procedure allows activities to be added to the new route or deleted from the current route while

the swap procedure encourages the changing of activities between routes. The combinations of both procedures allow the particle to explore the solution space more thoroughly and are very effective techniques in improving the solution quality of PSO.

## 5. COMPUTATIONAL EXPERIMENT

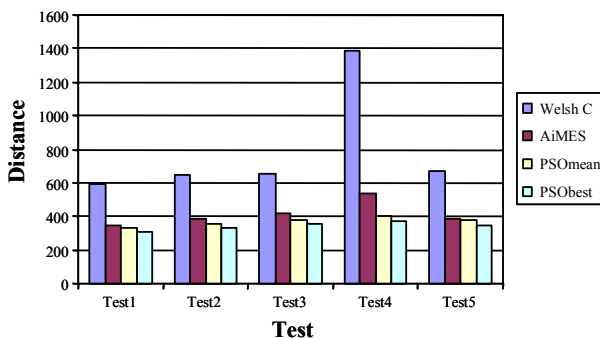
The algorithm has been coded in MATLAB and tests have been performed on a Pentium M processor computer, 1.6 GHz CPU speed, 512MB RAM.

To test the performance of the Improved PSO algorithm, a sample of 'real' data is used. There are five sets of test data selected randomly from the previous care service data. The data consists of over 100 activities per day required by 50 clients and carried out by 12 care workers.

The test results are compared with those obtained by the corresponding local government authority (the Welsh Systems Consortium) using its existing manual processes. They are also compared with the results obtained by the Advanced Internet and Emergent Systems Centre at the University of Liverpool, when using the proprietary software ILOG™ Dispatcher 4.0.

For each case, PSO was run for 20 replications. Table 4 shows the problem size of each case, a comparison of the results, and a comparison of percentage of savings. Figure 1 presents the graphical comparison of distance traveled of all cases. From the experimental results, it can be seen that the improved PSO yields consistently and substantially better results for all test cases. The percentage of savings of PSO<sub>best</sub> to Welsh Systems Consortium range from 91 to 274 percent, while the percentage of savings of PSO<sub>best</sub> to AiMES range from 11 to 44 percent.

To further verify the results, the statistical *t*-Test is



**Figure 1.** Comparison of distance traveled of Welsh Systems Consortium, AiMES, PSO<sub>mean</sub>, and PSO<sub>best</sub>.

applied to test the difference in mean of the objective function between Improved PSO and AiMES. MINITAB® Release 14 is used in the analysis of results. The results have shown that  $p$ -value of test cases Monday to Thursday are equal to 0 while  $p$ -value of Friday is equal to 0.017. Since  $p < 0.05$  for all test cases, then at the 95% confidence level the mean differences of all cases are significant. It can be concluded that the results obtained from Improved PSO are significantly better than those obtained from AiMES, at the 95% confidence level.

The average computation time for the improved PSO is approximately 3.5 minutes whereas ILOG averaged 0.5 minutes. However, such a comparison is not conclusive as the experimental PSO software has not been developed and tuned to commercial standards.

## 6. CONCLUSION AND FUTURE RESEARCH

This paper has presented an improved particle swarm optimization (PSO) algorithm for scheduling home care workers, with the objective of minimising the sum of the distances travelled by the care workers, without violating the capacity and delivery time window constraints. To the authors knowledge, this is the first time that a PSO metaheuristic has been applied to care worker scheduling.

Due to the continuous nature of PSO, the *heuristic assignment* scheme has been especially designed for this problem in order to translate the continuous value to a discrete schedule. The initial solution technique, *Earliest Start Time Priority with Minimum Distance Assignment* (ESTPMDA) has been employed to direct the search direction of the particles used in the algorithm. During the searching stage of PSO, the local improvement procedures (LIP) *insertion* and *swap* have been applied in order to further improve the quality of the solution.

The proposed methodology has been tested on 'real' data and the results compared with those obtained using the current manual procedures and those obtained using ILOG™. For the test cases the improved PSO algorithm consistently produced the best results.

Future research will extend to a multi-objective function to minimize the number of care workers and the total distance traveled. Also, experiments will be conducted to understand more about the effects of the PSO algorithm's parameters.

## ACKNOWLEDGEMENTS

This research is supported by the Royal Golden Jubilee Ph.D. Program of Thailand Research Fund, contract number PHD/0084/2544 and the University of Liverpool Management School.

## REFERENCES

- Allahverdi, A. and Al-Anzi, F. S. (2006), A PSO and a tabu search heuristics for the assembly scheduling problem of the two-stage distributed database application, *Computers and Operations Research*, **33**, 1056-1080.
- Bräysy, O., Dullaert, W., and Gendreau, M. (2004), Evolutionary algorithms for the vehicle routing problem with time windows, *Journal of Heuristics*, **10**, 587-611.
- Chiang, W. and Russell, R. (1996), Simulated annealing metaheuristics for the vehicle routing problem with time windows, *Annals of Operations Research*, **63**, 3-27.
- Clerc, M. (1999), The swarm and the queen: towards a deterministic and adaptive particle swarm optimization, *Proceeding of the IEEE Congress on Evolutionary Computation* (Washington DC), Piscataway, NJ IEEE service center, 1951-1957.
- Desrochers, M., Lentra, J., Savelsbergh, M., and Soumis, F. (1988), Vehicle routing with time windows: optimization and approximation, *Vehicle Routing: Methods and Studies*, North-Holland, Amsterdam, 64-84.
- Desrochers, M., Desrosiers, J., and Solomon, M. (1992), A new optimization algorithm for the vehicle routing problem with time windows, *Operations Research* **40**, 342-354.
- Eberhart, R. C. and Kennedy, J. (1995), New optimizer using particle swarm theory, *Proceeding of the Sixth International Symposium on Micro Machine and Human Science*, Piscataway, NJ IEEE service center, Nagoya, Japan, 39-43.
- Eberhart, R. C. and Shi, Y. (2001), Particle swarm optimization: developments, applications, and resources, *Proceedings of the 2001 Congress on Evolutionary Computation* (Seoul Korea), Piscataway, NJ IEEE service center, 81-86.
- Fisher, M. L., Jornsten, K. O., Madsen, O. B. G. (1997), Vehicle routing with time windows: two optimization algorithms, *Operations Research*, **45**, 488-492.
- Gambardella, L. M., Taillard, E., and Agazzi, G. (1999), MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows, *New Ideas in Optimization*, McGraw-Hill, (Also available as, *Technical Report IDSIA-06-99*, IDSIA, Lugano, Switzerland).
- Garcia, B. L., Potvin, J. Y., and Rousseau, J. M. (1994), A parallel implementation of the tabu search heuristic for vehicle routing problems with time window constraints, *Computers and Operations Research*, **21**, 1025-1033.
- Homberger, J. and Gehring, H. (2005), A two-phase hybrid metaheuristic for routing problems with time windows, *European Journal of Operational Research*, **162**, 220-238.

- Hwa, F., Liu, F., Shen, S. Y. (1999), A route-neighborhood-based metaheuristic for vehicle routing problem with time windows, *European Journal of Operational Research*, **128**, 485-504.
- Jerald, J., Asokan, P., Prabakaran, G., and Saravanan, R. (2005), Scheduling optimisation of flexible manufacturing systems using particle swarm optimisation algorithm, *International Journal of Advanced Manufacturing Technology*, **25**, 964-971.
- Kennedy, J. and Eberhart, R. C. (1995), Particle swarm optimization, *Proceeding of IEEE International Conference on Neural Networks IV*, Piscataway, NJ IEEE service center, 1942-1948.
- Kolen, A. W. J., Rinnooy, Kan A. H. G., and Trienekens, H. W. J. M. (1987), Vehicle routing with time windows, *Operations Research*, **35**, 266-273.
- Lin, S. and Kernighan, B. W. (1973), An effective heuristic algorithm for the Traveling Salesman problem, *Operations Research*, **21**, 498-516.
- Polacek, M., Hartl, R. F., Doerner, K., and Reimann, M. (2004), A variable neighborhood search for the multi depot vehicle routing problem with time windows, *Journal of Heuristics*, **10**(6), 613-627.
- Potvin, J. Y. and Rousseau, J. M. (1993), A parallel route building algorithm for the vehicle routing and scheduling problem with time window, *European Journal of Operational Research*, **66**, 331-340.
- Potvin, J. Y. and Rousseau, J. M. (1995), Exchange heuristic for routing problems with time windows, *Journal of the Operational Research Society*, **46**, 1433-1446.
- Potvin, J., Kervahut, T., Garcia, B., and Rousseau, J. (1996), The vehicle routing problem with time windows, part I: tabu search, *Inform Journal of Computing*, **8**, 158-164.
- Potvin, J. Y. and Bengio, S. (1996), The vehicle routing problem with time windows-partII: genetic search, *Inform Journal of Computing*, **8**, 165-172.
- Savelsbergh, M. W. P. (1990), An efficient implementation of local search algorithms for constrained routing problems, *European Journal of Operational Research*, **47**, 75-85.
- Shi, Y. and Eberhart, R. C. (1998), Parameter selection in particle swarm optimization, *Proceedings of the 7th international Conference on Evolutionary Programming VII*, New York, Springer-Verlag, 591-600.
- Solomon, M. M. (1987), Algorithms for the vehicle routing and scheduling problem with time window constraints, *Operations Research*, **35**, 254-265.
- Solomon, M. and Desrosiers, J. (1988), Time window constrained routing and scheduling problem, *Transportation Science*, **22**, 1-13.
- Taillard, E., Badeau, P., Gendreau, M., Guertin, F., and Potvin, J., (1997), A tabu search heuristic for the vehicle routing problem with soft time windows, *Transportation Science*, **31**, 170-186.
- Tan, K. C., Lee, L. H., Zhu, Q. L., and Ou, K. (2001), Heuristic methods for vehicle routing problem with time windows, *Artificial Intelligence in Engineering*, **15**, 281-295.
- Tasgetiren, M. F., Liang, Y. C., Sevkli, M., and Gencyilmaz, G. (2004), Particle swarm optimization algorithm for single machine total weighted tardiness problem, *IEEE*, 1412-1419.
- Tasgetiren, M. F., Liang, Y. C., Sevkli, M., and Gencyilmaz, G. (2007), Particle swarm optimization algorithm for makespan and total flowtime minimization in permutation flowshop sequencing problem, *European Journal of Operational Research*, **177**, 1930-1947.
- Thangiah, S. R., Osman, I. H., and Sun, T. (1994), Hybrid genetic algorithm, simulated annealing and tabu search methods for vehicle routing problems with time windows, *Technical report SRU-CpSc-TR-94-27, Computer Science Department, Slippery Rock University*.
- Xia, W. and Wu, Z. (2005), An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems, *Computers and Industrial Engineering*, **48**, 409-425.
- Zhang, H., Li, H., and Tam, C. M. (2006), Particle swarm optimization for resource-constrained project scheduling, *International Journal of Project Management*, **24**, 83-92.