Journal of Universal Computer Science, vol. 13, no. 9 (2007), 1309-1331
submitted: 12/6/06, accepted: 24/10/06, appeared: 28/9/07

An Hybrid Fuzzy Variable Neighborhood Particle Swarm Optimization Algorithm for Solving Quadratic Assignment Problems

Hongbo Liu
(Dalian Maritime University, China
liuhb@newmail.dlmu.edu.cn)

Ajith Abraham*
(Center of Excellence for Quantifiable Quality of Service
Norwegian University of Science and Technology,
Trondheim, Norway
ajith.abraham@ieee.org)

Abstract: Recently, Particle Swarm Optimization (PSO) algorithm has exhibited good performance across a wide range of application problems. A quick review of the literature reveals that research for solving the Quadratic Assignment Problem (QAP) using PSO approach has not much been investigated. In this paper, we design a hybrid meta-heuristic fuzzy scheme, called as variable neighborhood fuzzy particle swarm algorithm (VNPSO), based on fuzzy particle swarm optimization and variable neighborhood search to solve the QAP. In the hybrid fuzzy scheme, the representations of the position and velocity of the particles in the conventional PSO is extended from the real vectors to fuzzy matrices. A new mapping is introduced between the particles in the swarm and the problem space in an efficient way. We also attempt to theoretically prove that the variable neighborhood particle swarm algorithm converges with a probability of 1 towards the global optimal. The performance of the proposed approach is evaluated and compared with other four different algorithms. Empirical results illustrate that the approach can be applied for solving quadratic assignment problems effectively.

Key Words: Particle swarm optimization, Variable neighborhood search, Quadratic assignment problem

Category: I.2, I.2.2, I.2.8

1 Introduction

Particle Swarm Optimization (PSO) algorithm is inspired by social behavior patterns of organisms that live and interact within large groups. In particular, PSO incorporates swarming behaviors observed in flocks of birds, schools of fish, or swarms of bees, and even human social behavior, from which the Swarm Intelligence (SI) paradigm has emerged [Kennedy and Eberhart 2001, Clerc 2006]. It could be implemented and applied easily to solve various function optimization

*Corresponding author

problems, or the problems that can be transformed to function optimization problems [Parsopoulos and Vrahatis 2002, Margarita and Carlos 2006]. As an algorithm, the main strength of PSO is its fast convergence, which compares with other global optimization algorithms [Eberhart and Shi 1998, Settles et al. 2003, Boeringer and Werner 2004]. PSO has exhibited good performance across a wide range of applications [Schute and Groenwold 2005, Abraham et al. 2006a]. However, research on discrete problems, especially Quadratic Assignment Problem (QAP), has not been reported very much in the literature [Pang et al. 2004, Kennedy and Eberhart 1997, Abraham et al. 2006b]. In this paper, we design a hybrid meta-heuristic fuzzy scheme, called as VNPSO, based on discrete particle swarm optimization variable neighborhood search to solve quadratic assignment problems.

The rest of the paper is organized as follows. Related works about QAP is reviewed in Section 2. We formulate the quadratic assignment problem in Section 3. Particle swarm model is presented and the effects on the change of the velocities of particles are analyzed in Section 4. In Section 5, we describe the VNPSO model in detail. In Section 6, we theoretically prove the properties related to the convergence of the proposed algorithm. Experiment settings, results and discussions are given in Section 7 and finally conclusions are given in Section 8.

2 Related Works

Koopmans and Beckmann first proposed the quadratic assignment problem as a mathematical model related to economic activities [Koopmans et al. 1957]. Since then, it has several good practical applications in diverse areas [Heffley 1980], [Anstreicher et al. 2002, Ciriani et al. 2004, Miranda et al. 2005], [Loiola et al. 2007]. Since its first formulation, the QAP has been drawing researchers' attention worldwide, not only because of its practical and theoretical importance, but also because of its complexity. The QAP is a NP-hard optimization problem [Wikipedia 2006]. While some NP-hard combinatorial optimization problems can be solved exactly for relatively large instances, as exemplified by the traveling salesman problem (TSP), QAP instances of size larger than 20 are considered intractable.

Researchers worldwide made attempts to achieve a global optimum for the QAP using variant methods, including the exact, heuristic and hybrid methods. Gilmore [Gilmore 1962] and Lawler [Lawler 1963] presented firstly the enumerative schemes that use lower bounds to eliminate undesired solutions. Some variations of the approach are proposed in [Bozer and Suk-Chul 1996], [Ball et al. 1998]. Some methods that combine branch-and-bound with parallel techniques are used effectively in [Mautor and Roucairol 1994], [Clausen and Perregaard 1997]. Cutting plane methods and its formulation of some heuristics are introduced in [Bazaraa and Sherali 1982, Miranda et al. 2005] The branch-and-cut technique, a variation is proposed in [Jünger and Kaibel 2001]. Erdoğan and Tansel reports recently some computational experiences and results with a branch-and-cut algorithm [Erdoğan and Tansel 2007]. Since the QAP is one of the hardest optimization problems, it is difficult for the exact algorithms to exhibit a very good performance on the problem, specially on the large scale ones [Angel and Zissimopoulos 2000]. Therefore, several heuristics and meta-heuristics have been proposed for finding optimum or near-optimum solutions for the problems. Constructive methods are used in [Gutin and Yeo 2002], [Arkin et al. 2001], [Yu and Sarker 2003]. Arora et al. proposed a randomized procedure for rounding fractional perfect assignments to integral assignments [Arora et al. 2002]. Several general techniques, known as meta-heuristics, are based on some form of simulation of a natural process studied within another field of knowledge (metaphors). With the advent of meta-heuristics, QAP research received new and increased interest [Loiola et al. 2007]. Since Burkard and Rendl used Simulated Annealing (SA) to solve the QAP [Burkard and Rendl 1984], many improved methods based SA are presented in [Bos 1993, Tian et al. 1996, Misevicius 2003]. Tabu Search (TS) for the QAP can be found in [Chakrapani and Skorin-Kapov 1993], [Taillard 1991], [Drezner 2005], [Misevicius 2005]. Despite the inconvenience of depending on the size of the tabu list and the way which the list is managed, the performances of those algorithms show they are very efficient strategies for the QAP. More researchers investigate how to use Genetic Algorithms (GA) on QAP [Tate and Smith 1995, Kochhar et al. 1998, Lim et al. 1998, Drezner et al. 2003, El-Baz 2004, Drezner et al. 2005a]. There are some other search techniques for solving the QAP, such as scatter search [Cung et al. 1997], memetic algorithms [Merz and Freisleben 2000], greedy algorithm [Angel et al. 2000], greedy randomized adaptive search procedure [Oliveira et al. 2004], variable neighborhood search recently [Zhang et al. 2005], chaotic search [Hasegawa et al. 2002]. Some local search techniques are also used to achieve the solution to QAP [Mills et al. 2003, Angel and Zissimopoulos 1998, Stützle 2006]. Many research show Ant Colony Optimization (ACO) technique as a competitive meta-heuristic for the QAP [Gambardella et al. 1999], [Maniezzo and Colorni 1999], [Stützle and Holger 2000, Talbi et al. 2001], [Middendorf et al. 2002], [Randall 2004, Solimanpur et al. 2004, Wiesemann and Stützle 2006]. All the above methods possess respective advantages and disadvantages. Some of comparison research has been reported in [Taillard 1995], [Merz and Freisleben 1999]. However, as reported in the literature, some hybrid ideas has proven to be more efficient [Loiola et al. 2007]. Yip and Sarker proposed a guided evolutionary simulated annealing approach for solving the QAP [Yip and Pao 1994]. A combination of simulated annealing and genetic algorithm is presented in [Boülte and Thonemann 1996]. Battiti and Tecchiolli used tabu search with simulated annealing for solving the QAP [Battiti and Tecchiolli 1994]. Ostrowski and Ruoppila proposed hybrid genetic algorithms based on GA and SA [Ostrowski and Ruoppila 1997]. Some techniques used parallel and massive computation for solving QAP [Roucairol 1987, Chakrapani and Skorin-Kapov 1993, Talbi et al. 2001]. In addition, some assistant strategies are often used to explore a better solution for the QAP. Multistart techniques are usually used to begin heuristic or meta-heuristic methods [Fleurent and Glover 1999, Misevicius 2004]. Kelly et al. studied diversification strategies for the QAP [Kelly et al. 1994]. Nissen and Paul applied the threshold accepting technique to the QAP [Nissen and Paul 1995]. Most of the hybrid algorithms have proven to be more promising than the corresponding algorithms when used in a stand-alone mode. More survey of the related literature are reported in [Çela 1998, Anstreicher 2003, Drezner et al. 2005b, Loiola et al. 2007], which would be very helpful for the readers to understand the QAP more completely.

3 Quadratic Assignment Problem

The quadratic assignment problem is a standard problem in location theory. It was introduced by Koopmans and Beckmann in [Koopmans et al. 1957] and is a model for the practical problems [Fedjki and Duffuaa 2004, Stützle 2006]. Intuitively, the QAP can be described as the problem of assigning a set of facilities to a set of locations with given distances between the locations and given flows between the facilities. The goal then is to place the facilities on locations in such a way that the sum of the product between flows and distances is minimal. More formally, given n facilities $\{F_1, F_2, \cdots, F_n\}$ and n locations $\{L_1, L_2, \cdots, L_n\}$, two $n \times n$ matrices $FM = [f_{ij}]$ and $DM = [d_{rs}]$, where f_{ij} is the flow between facilities F_i and F_j and d_{rs} is the distance between locations L_r and L_s, the QAP can be stated as follows:

$$\min_{\Pi \in P(n)} Z_\Pi = \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} d_{\Pi_i \Pi_j} \tag{1}$$

where $P(n)$ is the set of all permutations (corresponding to the assignment solutions) of the set of integers $\{1, 2, \cdots, n\}$, and Π_i gives the location of facility F_i in the current solution $\Pi \in P(n)$. Here $f_{ij} d_{\Pi_i \Pi_j}$ describes the cost contribution of simultaneously assigning facility F_i to location Π_i and facility F_j to location Π_j. It is to be noted that the number of facilities (n) is assumed to be the same as the number of locations. In the other word, one facility could be assigned to only one location, and one location could be assigned to only one facility in a feasible assignment solution.

The term *quadratic* stems from the formulation of the QAP as an integer optimization problem with a quadratic objective function [Stützle 2006]. Let b_{ij}

be a binary variable which takes value 1 if facility F_i is assigned to location L_j and 0 otherwise. Then the problem can be re-formulated as:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{r=1}^{n} \sum_{s=1}^{n} f_{ij} d_{rs} b_{ir} b_{js} \tag{2}$$

s.t.

$$\sum_{i=1}^{n} b_{ij} = 1,\ i = 1, 2, \cdots, n,\ j = 1, 2, \cdots, n; \tag{3}$$

$$\sum_{j=1}^{n} b_{ij} = 1,\ i = 1, 2, \cdots, n,\ j = 1, 2, \cdots, n; \tag{4}$$

$$b_{ij} \in \{0, 1\},\ i = 1, 2, \cdots, n,\ j = 1, 2, \cdots, n. \tag{5}$$

4 Particle Swarm Model

The classical particle swarm model consists of a swarm of particles, which are initialized with a population of random candidate solutions. They move iteratively through the d-dimension problem space to search the new solutions, where the fitness, f, can be calculated as the certain qualities measure. Each particle has a position represented by a position-vector $\mathbf{p}_i$ (i is the index of the particle), and a velocity represented by a velocity-vector $\mathbf{v}_i$. Each particle remembers its own best position so far in a vector $\mathbf{p}_i^{\#}$, and its j-th dimensional value is $p_{ij}^{\#}$. The best position-vector among the swarm so far is then stored in a vector $\mathbf{p}^*$, and its j-th dimensional value is p_j^*. During the iteration time t, the update of the velocity from the previous velocity to the new velocity is determined by Eq. (6). The new position is then determined by the sum of the previous position and the new velocity by Eq. (7).

$$v_{ij}(t) = w v_{ij}(t-1) + c_1 r_1 (p_{ij}^{\#}(t-1) - p_{ij}(t-1)) + c_2 r_2 (p_j^*(t-1) - p_{ij}(t-1)) \tag{6}$$

$$p_{ij}(t) = p_{ij}(t-1) + v_{ij}(t) \tag{7}$$

where r_1 and r_2 are the random numbers in the interval [0,1]. c_1 is a positive constant, called as coefficient of the self-recognition component, c_2 is a positive constant, called as coefficient of the social component. The variable w is called as the inertia factor, which value is typically setup to vary linearly from 1 to near 0 during the iterated processing. From Eq.(6), a particle decides where to move next, considering its current state, its own experience, which is the memory of its best past position, and the experience of its most successful particle in the swarm.

In the particle swarm model, the particle searches the solutions in the problem space within a range $[-s, s]$ (If the range is not symmetrical, it can be translated to the corresponding symmetrical range.) In order to guide the particles effectively in the search space, the maximum moving distance during one iteration is clamped in between the maximum velocity $[-v_{max}, v_{max}]$ given in Eq. (8), and similarly for its moving range given in Eq. (9):

$$v_{ij} = sign(v_{ij})min(|v_{ij}|, v_{max}) \tag{8}$$

$$p_{ij} = sign(p_{ij})min(|p_{ij}|, p_{max}) \tag{9}$$

The value of v_{max} is $\rho \times s$, with $0.1 \leq \rho \leq 1.0$ and is usually chosen to be s, i.e. $\rho = 1$. The pseudo-code for particle swarm optimization algorithm is illustrated in Algorithm 1.

Algorithm 1 Particle Swarm Optimization Algorithm

```
Initialize the size of the particle swarm n, and other parameters.
Initialize the positions and the velocities for all the particles randomly.
While (the end criterion is not met) do
   t = t + 1;
   Calculate the fitness value of each particle;
   p* = argmin_{i=1}^{n}(f(p*(t-1)), f(p_1(t)), f(p_2(t)), ..., f(p_i(t)), ...,
    f(p_n(t)));
   For i= 1 to n
      p_i^#(t) = argmin_{i=1}^{n}(f(p_i^#(t-1)), f(p_i(t));
      For j = 1 to d
         Update the j-th dimension value of p_i and v_i
            according to Eqs.(6),(8),(7),(9);
      Next j
   Next i
End While.
```

Some previous studies have discussed the trajectory of particles and its convergence [Cristian 2003], [Van den Bergh and Engelbrecht 2006], [Liu et al. in press a], [Clerc and Kennedy 2002]. It has been shown that the trajectories of the particles oscillate as different sinusoidal waves and converge quickly, sometimes prematurely. The effects of the change in the velocities of particles has been paid little attention. The gradual change of the particle's velocity can be explained geometrically. During each iteration, the particle is attracted towards the location of the best fitness achieved so far by the particle itself and by the location of the best fitness achieved so far across the whole swarm. From Eq.(6), v_{ij} can attain a smaller value, but if the second term and the third term in RHS of Eq.(6) are both small, it cannot resume a larger value and could eventually loose the exploration capabilities in the future iterations. Such situations could occur even in the early stages of the search. When the second term and the third term in RHS of Eq.(6) are zero, v_{ij} will be damped quickly with the ratio of w. In other words, if a particle's current position coincides with the global best position/particle, the particle will only move away from this point if its previous velocity and w are non-zero. If their previous velocities are very close to zero, then all the particles will stop moving once they catch up with the global best particle, which many lead to premature convergence. In fact, this does not even guarantee that the algorithm has converged to a local minimum and it merely means that all the particles have converged to the best position discovered so far by the swarm. This state owes to the second term and the third term in the RHS of Eq.(6), the cognitive components of the PSO. But if the cognitive components of the PSO algorithm are invalidated, all particles always search the solutions using the initial velocities. Then the algorithm is merely a degenerative stochastic search without the characteristics of PSO. Some related research and development during the recent years are reported in [Grosan et al. 2005, Jiang et al. 2005, Liu et al. 2006a, Liu et al. 2006b, Liang et al. 2006].

5 Fuzzy Variable Neighborhood Particle Swarm Optimization for the Quadratic Assignment Problem

For applying the particle swarm algorithm successfully for any problem, one of the key issues is how to map the problem solution to the particle space, which affects its feasibility and performance [Salman et al. 2001, Pang et al. 2004]. In a "crisp" particle swarm model for the assignment problem, it would trend to assign many facilities to the same location or assign many locations to the same facility. Usually this kind of the assignment would be unfeasible. In this section, a fuzzy matrix is introduced to represent the quadratic assignment problem. Then, a new approach to the problem space mapping is depicted for particle swarm optimization with reference to the quadratic assignment problem. For solving the QAP, which is considered as one of the hardest optimization problems, it would be very difficult for a particle swarm search to use a simple exploring strategy to search the optimal solution. We introduce a novel shaking strategy to drive those "lazy" particles in the swarm and let them explore variable neighborhoods for better solutions.

5.1 A Fuzzy Particle Swarm Approach for QAP

Suppose $F = \{F_1, F_2, \cdots, F_n\}$, $L = \{L_1, L_2, \cdots, L_n\}$, then the fuzzy assignment relation from F to L can be expressed as follows:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

Here a_{ij} represents the degree of membership of the j-th element F_j in domain F and the i-th element L_i in domain L to relation A. In the fuzzy relation matrix A between F and L, the elements subject to the following constraints:

$$a_{ij} = \mu_R(F_j, L_i),\ i = 1, 2, \cdots, n,\ j = 1, 2, \cdots, n. \tag{10}$$

μ_R is the membership function, the value of a_{ij} means the degree of membership that the facility F_j would be assigned to the location L_i in the feasible assignment solution. In the quadratic assignment problem, the elements of the solution must satisfy the following conditions:

$$a_{ij} \in \{0, 1\},\ i = 1, 2, \cdots, n,\ j = 1, 2, \cdots, n; \tag{11}$$

$$\sum_{i=1}^{n} a_{ij} = 1,\ i = 1, 2, \cdots, n,\ j = 1, 2, \cdots, n; \tag{12}$$

$$\sum_{j=1}^{n} a_{ij} = 1,\ i = 1, 2, \cdots, n,\ j = 1, 2, \cdots, n. \tag{13}$$

According to the fuzzy matrix representation of the quadratic assignment problem, the position $\mathbf{p}$ and velocity $\mathbf{v}$ in the particle swarm are re-defined as follows:

$$\mathbf{p} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix}; \qquad \mathbf{v} = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ v_{n1} & v_{n2} & \cdots & v_{nn} \end{bmatrix}$$

The elements in the matrix $\mathbf{p}$ above have the same meaning as Eq.(10). Accordingly, the elements of the matrix $\mathbf{p}$ must satisfy the following conditions:

$$p_{ij} \in \{0, 1\},\ i = 1, 2, \cdots, n,\ j = 1, 2, \cdots, n; \tag{14}$$

$$\sum_{i=1}^{n} p_{ij} = 1,\ i = 1, 2, \cdots, n,\ j = 1, 2, \cdots, n; \tag{15}$$

$$\sum_{j=1}^{n} p_{ij} = 1,\ i = 1, 2, \cdots, n,\ j = 1, 2, \cdots, n. \tag{16}$$

Because the position and velocity in the new fuzzy particle swarm model have been transformed to the form of matrices, they are updated by the new Eqs.(17) and (18) with the matrix operations.

$$\begin{aligned}\mathbf{v}(t) = w \otimes \mathbf{v}(t-1) \oplus (c_1 r_1) \otimes (\mathbf{p}^{\#}(t-1) \ominus \mathbf{p}(t-1)) \\ \oplus (c_2 r_2) \otimes (\mathbf{p}^{*}(t-1) \ominus \mathbf{p}(t-1))\end{aligned} \tag{17}$$

$$\mathbf{p}(t) = \mathbf{p}(t-1) \oplus \mathbf{v}(t) \tag{18}$$

The position matrix may violate the constraints (14), (15) and (16) after some iterations, it is necessary to normalize the position matrix. First we make all the negative elements in the matrix to become zero. If all elements in a column of the matrix are zero, they need be re-evaluated using a series of random numbers within the interval [0,1]. And then the matrix undergoes the following transformation:

$$\mathbf{p_{normal}} = \begin{bmatrix} p_{11}/\sum_{i=1}^{n} p_{i1} & p_{12}/\sum_{i=1}^{n} p_{i2} & \cdots & p_{1n}/\sum_{i=1}^{n} p_{in} \\ p_{21}/\sum_{i=1}^{n} p_{i1} & p_{22}/\sum_{i=1}^{n} p_{i2} & \cdots & p_{2n}/\sum_{i=1}^{n} p_{in} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1}/\sum_{i=1}^{n} p_{i1} & p_{n2}/\sum_{i=1}^{n} p_{i2} & \cdots & p_{nn}/\sum_{i=1}^{n} p_{in} \end{bmatrix}$$

Since the position matrix indicates the potential assigned solution, the fuzzy matrix can be "decoded" to the feasible solution. We choose the element which has the max value in the column, then tag it as "1", and other numbers in the column and row are set as "0" in the assigning matrix. After all the columns and rows have been processed, we get the assignment solution without violating the constraints (14), (15) and (16), and then calculate the assignment cost of the solution. Although the fuzzy coding solution is a very good mapping idea, it is difficult for the particle swarm algorithm alone to search the best solutions to the QAP, especially for the large scale problems [Liu et al. in press b]. We will discuss a hybrid approach to the QAP in the next subsections.

5.2 Variable Neighborhood Particle Swarm Optimization for QAP

Variable Neighborhood Search (VNS) is a relatively recent metaheuristic which relies on iteratively exploring neighborhoods of growing size to identify better local optima with shaking strategies [Hansen and Mladenović 2003]. More precisely, VNS escapes from the current local minimum $\mathbf{x}^*$ by initiating other local searches from starting points sampled from a neighborhood of $\mathbf{x}^*$, which increases its size iteratively until a local minimum is better than the current one is found. These steps are repeated until a given termination condition is met. The shaking strategy usually is based on *insert* and *interchange* operators [Hansen and Mladenović 2001]. But for quadratic assignment problems, those operators would change the sequence of the encoded facilities, which makes it difficult to decode the particles' position to the potential solutions. The meta-heuristic method, the VNPSO, was originally inspired by VNS. In the particle swarm, if a particle's velocity decreases to a threshold v_c, a new velocity is assigned using Eq.(19):

$$p_{ij}(t) = w\hat{v} + c_1 r_1 (p_{ij}^{\#}(t-1) - p_{ij}(t-1)) + c_2 r_2 (p_j^*(t-1) - p_{ij}(t-1)) \quad (19)$$

$$\hat{v} = \begin{cases} v_{ij} & \text{if } |v_{ij}| \geq v_c \\ rand(-1,1)v_{max}/\eta & \text{if } |v_{ij}| < v_c \end{cases} \quad (20)$$

where $rand(-1,1)$ is the random number, normally distributed within the interval [-1,1], and η is the variable neighborhood scaling factor to control the domain of the particle's oscillation according to v_{max}. v_c is the minimum velocity threshold, a tunable threshold parameter to limit the minimum of the particles' velocity. The proposed algorithm scheme is summarized as Algorithm 2. For QAP, the variables and the operators need be transformed to the form of matrices, as illustrate Eqs. (17) and (18). The performance of the algorithm is directly correlated to two parameter values, v_c and η. A large v_c shortens the oscillation period, and it provides a great probability for the particles to leap over local minima using the same number of iterations. But a large v_c compels the particles in the quick "flying" state, which leads them not to search the solution and forcing them not to refine the search. The value of η changes directly the variable search neighborhoods for the particles. It is to be noted that the algorithm is different from the multi-start technique and the turbulence strategy. We also implemented the Multi-Start PSO (MSPSO) and Velocity Turbulent PSO (VTPSO) to compare their performances. To compare with other algorithms, the approach has the following differences:

- Our approach does not make a perturbation on the positions of the particles directly, while it takes a minimum velocity threshold to limit the particles' velocities. To do so, it avoids how to decide which one position should be made a perturbation. In fact, it is very difficult to decide which one particle's position need a perturbation during the particle swarm searching for the QAP.

- In the proposed algorithm, the particle is not on the run incessantly with a high velocity, which is larger than the minimum velocity, from beginning to end, even they are very close to the global optima, like them in VTPSO. They have the opportunities with 10 times in a variable neighborhood search loop to approach to the global optima with a low velocity, which is less than a minimum velocity threshold.

– The multi-start technique usually makes all the particles in the swarm not only forget its own experience but also forget the experience of the swarm's experience. It only remembers the best fitness value so far achieved. In the proposed approach, only the "lazy" particles are driven into a new neighborhood to explore the better solutions while the whole swarm is alive and iterated.

Algorithm 2 Variable Neighborhood Particle Swarm Optimization

01. Initialize the size of the particle swarm n, and other parameters.
02. Initialize the positions and the velocities for all the particles randomly.
03. Set the flag of iterations without improvement $Nohope = 0$.
04. While (the end criterion is not met) do
05. $t = t + 1$;
06. Calculate the fitness value of each particle;
07. $\mathbf{p}^* = argmin_{i=1}^{n}(f(\mathbf{p}^*(t-1)), f(\mathbf{p}_1(t)), f(\mathbf{p}_2(t)), \cdots, f(\mathbf{p}_i(t)), \cdots, f(\mathbf{p}_n(t)))$;
08. If $\mathbf{p}^*$ is improved then $Nohope = 0$, else $Nohope = Nohope + 1$.
09. For i= 1 to n
10. $\mathbf{p}_i^{\#}(t) = argmin_{i=1}^{n}(f(\mathbf{p}_i^{\#}(t-1)), f(\mathbf{p}_i(t))$;
11. For $j = 1$ to d
12. If $Nohope < 10$ then
13. Update the j-th dimension value of $\mathbf{p}_i$ and $\mathbf{v}_i$
13. according to Eqs.(6),(8),(7),(9);
14. else
15. Update the j-th dimension value of $\mathbf{p}_i$ and $\mathbf{v}_i$
15. according to Eqs.(20),(19),(7),(9).
16. Next j
17. Next i
18. End While.

6 Convergence Analysis of VNPSO

For analyzing the convergence of the proposed algorithm, we first introduce the definitions and lemmas [Guo and Tang 2001, He et al. 2005, Weisstein 2006], and then theoretically prove that the proposed variable neighborhood particle swarm algorithm converges with a probability 1 or strongly towards the global optimal.

Consider the problem (P) as

$$(P) = \begin{cases} minf(\mathbf{x}) \\ \mathbf{x} \in \Omega = [-s, s]^n \end{cases} \tag{21}$$

where $\mathbf{x} = (x_1, x_2, \cdots, x_n)^T$. $\mathbf{x}^*$ is the global optimal solution to the problem (P), let $f^* = f(\mathbf{x}^*)$. Let

$$D_0 = \{\mathbf{x} \in \Omega | f(\mathbf{x}) - f^* < \varepsilon\} \tag{22}$$
$$D_1 = \Omega \setminus D_0$$

for every $\varepsilon > 0$.

Assume that the i-th dimensional value of the particle's velocity decreases to a threshold v_c, then the shaking strategy is activated, and a variable neighborhood velocity is generated by Eq.(20). In $rand(-1,1)v_{max}/\eta$, $rand(-1,1)$ is a normal distributed random number within the interval [-1,1], and the scaling factor η is a positive constant to control the domain of the particle's oscillation according to v_{max}. Therefore the variable neighborhood velocity $\hat{v}$ belongs to the normal distribution. If $v_{max} = s$, then $\hat{v} \sim [-\frac{s}{\eta}, \frac{s}{\eta}]$. During the iterated procedure from the time t to $t+1$, let q_{ij} denote that $\mathbf{x}(t) \in D_i$ and $\mathbf{x}(t+1) \in D_j$. Accordingly the particles' positions in the swarm could be classified into four states: q_{00}, q_{01}, q_{10} and q_{01}. Obviously $q_{00} + q_{01} = 1$, $q_{10} + q_{11} = 1$.

Definition 1 Convergence in terms of probability. Let ξ_n a sequence of random variables, and ξ a random variable, and all of them are defined on the same probability space. The sequence ξ_n converges with a probability of ξ if

$$\lim_{n \to \infty} P(|\xi_n - \xi| < \varepsilon) = 1 \tag{23}$$

for every $\varepsilon > 0$.

Definition 2 Convergence with a probability of 1. Let ξ_n a sequence of random variables, and ξ a random variable, and all of them are defined on the same probability space. The sequence ξ_n converges almost surely or almost everywhere or with probability of 1 or strongly towards ξ if

$$P\left(\lim_{n \to \infty} \xi_n = \xi\right) = 1; \tag{24}$$

or

$$P\left(\bigcap_{n=1}^{\infty} \bigcup_{k \geq n} [|\xi_n - \xi| \geq \varepsilon]\right) = 0 \tag{25}$$

for every $\varepsilon > 0$.

Lemma 3 Borel-Cantelli Lemma. *Let $\{A_k\}_{k=1}^{\infty}$ be a sequence of events occurring with a certain probability distribution, and let A be the event consisting of the occurrences of a finite number of events A_k for $k = 1, 2, \cdots$. Then*

$$P\left(\bigcap_{n=1}^{\infty} \bigcup_{k \geq n} A_k\right) = 0 \tag{26}$$

if

$$\sum_{n=1}^{\infty} P(A_n) < \infty; \tag{27}$$

$$P\left(\bigcap_{n=1}^{\infty} \bigcup_{k\geq n} A_k\right) = 1 \tag{28}$$

if the events are totally independent and

$$\sum_{n=1}^{\infty} P(A_n) = \infty. \tag{29}$$

Lemma 4 Particle State Transference. *$q_{01} = 0$; $q_{00} = 1$; $q_{11} \leq c \in (0, 1)$ and $q_{10} \geq 1 - c \in (0, 1)$.*

Proof. In the proposed algorithm, the best solution is updated and saved during the whole iterated procedure. So $q_{01} = 0$ and $q_{00} = 1$.

Let $\hat{\mathbf{x}}$ is the position with the best fitness among the swarm so far as the time t, i.e. $\hat{\mathbf{x}} = \mathbf{p}^*$. As the definition in Eq. (22), $\exists r > 0$, when $\|\mathbf{x} - \hat{\mathbf{x}}\|_\infty \leq r$, we have $|f(\mathbf{x}) - f^*| < \varepsilon$. Denote $Q_{\hat{\mathbf{x}},r} = \{x \in \Omega | \|\mathbf{x} - \hat{\mathbf{x}}\|_\infty \leq r\}$. Accordingly

$$Q_{\hat{\mathbf{x}},r} \subset D_0 \tag{30}$$

Then,

$$\begin{aligned} P\{(\mathbf{x} + \Delta\mathbf{x}) \in Q_{\hat{\mathbf{x}},r}\} &= \prod_{i=1}^{n} P\{|x_i + \Delta x_i - \hat{x}_i| \leq r\} \\ &= \prod_{i=1}^{n} P\{\hat{x}_i - x_i - r \leq \Delta x_i \leq \hat{x}_i - x_i + r\} \end{aligned} \tag{31}$$

where x_i, Δx_i and $\hat{x}_i$ are the i-th dimensional values of $\mathbf{x}$, $\Delta\mathbf{x}$ and $\hat{\mathbf{x}}$, respectively. Moreover, $\hat{\upsilon} \sim [-\frac{s}{\eta}, \frac{s}{\eta}]$, so that

$$P((\mathbf{x} + \Delta\mathbf{x}) \in Q_{\hat{\mathbf{x}},r}) = \prod_{i=1}^{n} \int_{\hat{x}_i - x_i - r}^{\hat{x}_i - x_i + r} \frac{\eta}{2\sqrt{2\pi}s} e^{-\frac{\eta^2 y^2}{2s^2}} dy \tag{32}$$

Denote $P_1(\mathbf{x}) = P\{(\mathbf{x} + \Delta\mathbf{x}) \in Q_{\hat{\mathbf{x}},r}\}$ and $\mathbb{C}$ is the convex closure of level set for the initial particle swarm. According to Eq. (32), $0 < P_1(\mathbf{x}) < 1$ ($\mathbf{x} \in \mathbb{C}$). Again, since $\mathbb{C}$ is a bounded closed set, so $\exists\hat{\mathbf{y}} \in \mathbb{C}$,

$$P_1(\hat{\mathbf{y}}) = \min_{\mathbf{x}\in\mathbb{C}} P_1(\mathbf{x}), \quad 0 < P_1(\hat{\mathbf{y}}) < 1. \tag{33}$$

Considering synthetically Eqs. (30) and (33), so that

$$q_{10} \geq P_1(\mathbf{x}) \geq P_1(\hat{\mathbf{y}}) \tag{34}$$

Let $c = 1 - P_1(\hat{\mathbf{y}})$, thus,

$$q_{11} = 1 - q_{10} \leq 1 - P_1(\hat{\mathbf{y}}) = c \quad (0 < c < 1) \tag{35}$$

and

$$q_{10} \geq 1 - c \in (0, 1) \tag{36}$$

Theorem 5. *Assume that the VNPSO algorithm provides position series $\mathbf{p}_i(t)$ $(i = 1, 2, \cdots, n)$ at time t by the iterated procedure. $\mathbf{p}^*$ is the best position among the swarm explored so far, i.e.*

$$\mathbf{p}^*(t) = arg \min_{1 \leq i \leq n} (f(\mathbf{p}^*(t-1)), f(\mathbf{p}_i(t))) \tag{37}$$

Then,

$$P\left(\lim_{t \to \infty} f(\mathbf{p}^*(t)) = f^*\right) = 1 \tag{38}$$

Proof. For $\forall \varepsilon > 0$, let $p_k = P\{|f(\mathbf{p}^*(k)) - f^*| \geq \varepsilon\}$, then

$$p_k = \begin{cases} 0 & \text{if } \exists T \in \{1, 2, \cdots, k\},\ \mathbf{p}^*(T) \in D_0 \\ \bar{p}_k & \text{if } \mathbf{p}^*(t) \notin D_0,\ t = 1, 2, \cdots, k \end{cases} \tag{39}$$

According to Lemma 4,

$$\bar{p}_k = P\{\mathbf{p}^*(t) \notin D_0, t = 1, 2, \cdots, k\} = q_{11}^k \leq c^k. \tag{40}$$

Hence,

$$\sum_{k=1}^{\infty} p_k \leq \sum_{k=1}^{\infty} c^k = \frac{c}{1-c} < \infty. \tag{41}$$

According to Lemma 3,

$$P\left(\bigcap_{t=1}^{\infty} \bigcup_{k \geq t} |f(\mathbf{p}^*(k)) - f^*| \geq \varepsilon\right) = 0 \tag{42}$$

As defined in Definition 2, the sequence $f(\mathbf{p}^*(t))$ converges almost surely or almost everywhere or with probability 1 or strongly towards f^*. The theorem is proven.

7 Experiments to Illustrate Algorithm Performance

In our experiments, Multi-Start PSO (MSPSO), Velocity Turbulent PSO (VTPSO), Multi-start Genetic Algorithm (MSGA) and Multi-start Ant Colony Optimization (MSACO) was used to compare the performance with Variable Neighborhood PSO (VNPSO). The GA and ACO algorithms share many similarities, which are also inspired from the nature like the PSO. GA mimic an evolutionary natural selection process. Generations of solutions are evaluated according to a fitness value and only those candidates with high fitness values are used to create further solutions via crossover and mutation procedures [Abraham et al. 2006c]. ACO deals with artificial systems that is inspired from the foraging behavior of real ants, which are used to solve discrete optimization problems [Dorigo and Stützle 2004]. The main idea is the indirect communication between the ants by means of chemical pheromone trials, which enables them to find short paths between their nest and food. It is implemented as a team of intelligent agents which simulate the ants behavior, walking around the graph representing the problem to solve using mechanisms of cooperation and adaptation. PSO is a stochastic search technique inspired by social behavior of bird flocking or fish schooling. All of them are valid and efficient methods in numeric programming and have been employed in various fields due to their strong convergence properties. Specific parameter settings for the algorithms are described in Table 1. We consider the instances from Taillard's datasets[1] and QAPlib[2]. Each experiment (for each algorithm) was repeated 5 times with different random seeds. Each trial had a fixed number of $50 * n * n$ iterations (n is the dimension of the problem). If the value $50 * n * n$ is larger than $2 * 10^4$, the maximum iteration was set to $2 * 10^4$. The average costs were calculated from the 5 different trials. Usually the main emphasis will be to generate the assignment solutions at a minimal amount of time. So the completion time for 5 trials were used as one of the criteria to improve their performance.

In order to closely track the performance of our algorithms, we tested the problem, tai12a. The tai12a is a QAP instance with 12 facilities on 12 locations. Its united matrix DF of the distance and flow is

$$\begin{array}{cc}
 & \begin{array}{cccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \end{array} \\
\begin{array}{c} 1\\2\\3\\4\\5\\6\\7\\8\\9\\10\\11\\12 \end{array} &
\begin{pmatrix}
0 & 21 & 95 & 82 & 56 & 41 & 6 & 25 & 10 & 4 & 63 & 6 \\
27 & 0 & 44 & 40 & 75 & 79 & 0 & 89 & 35 & 9 & 1 & 85 \\
85 & 80 & 0 & 84 & 12 & 0 & 26 & 91 & 11 & 35 & 82 & 26 \\
2 & 58 & 3 & 0 & 69 & 56 & 86 & 45 & 91 & 59 & 18 & 76 \\
1 & 21 & 48 & 74 & 0 & 39 & 18 & 57 & 36 & 61 & 36 & 21 \\
15 & 76 & 29 & 45 & 77 & 0 & 71 & 11 & 29 & 82 & 82 & 6 \\
11 & 72 & 90 & 65 & 36 & 91 & 0 & 71 & 8 & 77 & 74 & 30 \\
35 & 44 & 66 & 40 & 53 & 13 & 87 & 0 & 89 & 76 & 76 & 40 \\
11 & 85 & 41 & 54 & 37 & 29 & 67 & 10 & 0 & 93 & 56 & 1 \\
20 & 94 & 15 & 83 & 26 & 11 & 94 & 99 & 99 & 0 & 50 & 4 \\
21 & 90 & 83 & 14 & 87 & 77 & 79 & 56 & 60 & 56 & 0 & 36 \\
61 & 51 & 96 & 71 & 76 & 32 & 2 & 70 & 4 & 2 & 60 & 0
\end{pmatrix}
\end{array}$$

Figure 1 illustrates the performance of the five algorithms for tai12a. The results

[1] http://ina2.eivd.ch/collaborateurs/etd/
[2] http://www.opt.math.tu-graz.ac.at/qaplib/

Table 1: Parameter settings for the algorithms.

Algorithm	Parameter name	Parameter value
GA	size of the population	10
	Probability of crossover	0.8
	Probability of mutation	0.07
ACO	Number of ants	10
	Weight of pheromone trail α	1
	Weight of heuristic information β	5
	Pheromone evaporation parameter ρ	0.8
	Constant for pheromone updating Q	10
PSOs	Swarm size	10
	Self-recognition coefficient c_1	1.49
	Social coefficient c_2	1.49
	Inertia weight w	$0.9 \rightarrow 0.1$

of 5 MSGA runs were {249048, 252020, 244784, 249448, 252014}, with an average value of 249460, and assigning time of 897.8910. The results of 5 MSACO runs were {250788, 247440, 251754, 252826, 255902}, with an average value of 251742, and assigning time of 1362.0. The results of 5 MSPSO runs were {250394, 239412, 247486, 246084, 234596}, with an average value of 243590, and assigning time of 811.3750. The results of 5 VTPSO runs were {236822, 242412, 245510, 249176, 241624}, with an average value of 243110, and assigning time of 898.9220. The results of 5 VNPSO runs were {233040, 235704, 235776, 244316, 248150}, with an average value of 239400, and assigning time of 836.3750. As depicted by these results and those results in [Liu et al. in press b], the ACO and MSACO is effective algorithms for small scale problems, while VNPSO usually had better averages for a bigger problem sizes. Although the results of MSACO are not very good than other algorithms, its robustness is the best one among the considered algorithms. MSPSO usually spent the lowest time to assign the facilities on the locations, while MSACO spent the longest time to do that. MSGA does not provide the satisfied results, but it might be possible to achieve better solutions with a larger size of the population. This would also result in a longer assigning time.

8 Conclusions

Particle swarm optimization algorithm has exhibited good performance across a wide range of real world applications but not much work has been reported of its usage to solve the class of quadratic assignment problems. Initial difficulty

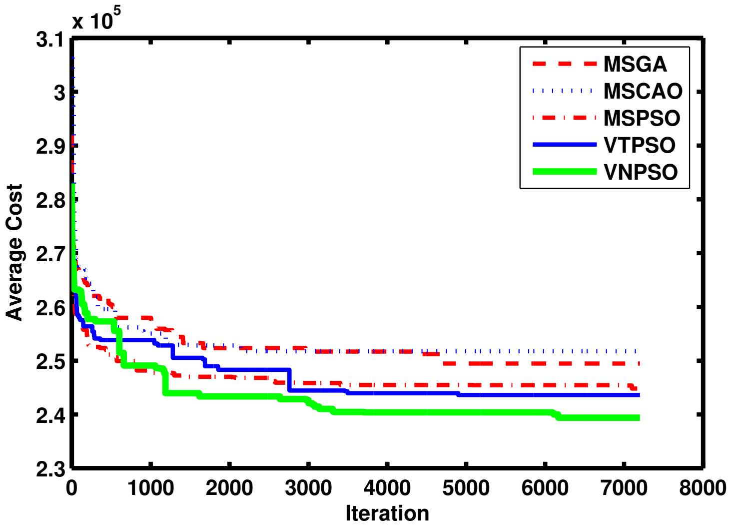

Figure 1: Performance for tai12a

consists of how to map the problem solution to the particle space. In this paper, we extend the representations of the position and velocity of the particles in PSO from the real vectors to fuzzy matrices, through which we accomplished the mapping between the quadratic assignment problem and the particle. We applied the new mapping method to design a hybrid fuzzy scheme, called as variable neighborhood fuzzy particle swarm algorithm (VNPSO), based on discrete particle swarm optimization and variable neighborhood search to achieve the global optimum for the quadratic assignment problems. The proposed VNPSO algorithm is illustrated theoretically that it converges with a probability of 1 towards the global optimum. We evaluated the performance of our proposed approach and compared it with Multi-Start PSO, Velocity Turbulent PSO, Multi-start Genetic Algorithm and Multi-start Ant Colony Optimization. Empirical results illustrated that the proposed approach was an effective approach to solve the quadratic assignment problems.

Acknowledgements

We would like to thank Drs. Bo Li, Ran He, Conghui Guo and Prof. Xiukun Wang for their scientific collaboration in this research work. This work is supported partly by NSFC (60573087), MOST (2005CB321904), and MOE (KP0302). Ajith Abraham acknowledges the support by the Centre for Quantifiable Quality of Service in Communication Systems, Centre of Excellence, appointed by The Research Council of Norway, and funded by the Research Council, NTNU and

UNINETT.

References

[Abraham et al. 2006a] Abraham A., Guo H., Liu H.: "Swarm Intelligence: Foundations, Perspectives and Applications"; "Swarm Intelligent Systems, Studies in Computational Intelligence, Chapter 1" (Nedjah, N., Mourelle, L. eds.), Springer (2006), 3-25

[Abraham et al. 2006b] Abraham A., Liu H., Zhang W.: "Scheduling Job on Computational Grids Using Fuzzy Particle Swarm Algorithm"; "Proceedings of the Tenth International Conference on Knowledge-Based & Intelligent Information & Engineering Systems, KES 2006", Lecture Notes in Artificial Intelligence, 4252(2006), 500-507

[Abraham et al. 2006c] Abraham A., Nedjah N., Mourelle L.: "Evolutionary Computation: From Genetic Algorithms to Genetic Programming"; "Studies in Computational Intelligence" (Nedjah N. et al. eds.), Springer (2006), 1-20

[Angel et al. 2000] Ahuja R., Orlin J. B., Tiwari A.: "A Greedy Genetic Algorithm for the Quadratic Assignment Problem"; Computers and Operations Research, 27, 10(2000), 917-934

[Angel and Zissimopoulos 1998] Angel E., Zissimopoulos V.: "On the Quality of Local Search for the Quadratic Assignment Problem"; Discrete Applied Mathematics, 82, 1-3(1998), 15-25

[Angel and Zissimopoulos 2000] Angel E., Zissimopoulos V.: "On the Hardness of the Quadratic Assignment Problem with Metaheuristics"; Journal of Heuristics, 8(2002), 399-414

[Anstreicher et al. 2002] Anstreicher K., Brixius N., Goux J.-P., Linderoth J.: "Solving Large Quadratic Assignment Problems on Computational Grids"; Mathematical Programming, Serial B, 91(2002), 563-588

[Anstreicher 2003] Anstreicher K. M.: "Recent Advances in the Solution of Quadratic Assignment Problems"; Mathematical Programming, 97, 1-2(2003), 27-42

[Arkin et al. 2001] Arkin E. M., Hassin R., Sviridenko M.: "Approximating the Maximum Quadratic Assignment Problem"; Information Processing Letters, 77, 1(2001), 13-16

[Arora et al. 2002] Arora S., Frieze A., Kaplan H.: "A New Rounding Procedure for the Assignment Problem with Applications to Dense Graph Arrangement Problems"; Mathematical Programming, 92, 1 (2002), 1-36.

[Ball et al. 1998] Ball M.O., Kaku B. K., Vakhutinsky A.: "Network-based Formulations of the Quadratic Assignment Problem"; European Journal of Operational Research, 104, 1(1998), 241-249

[Battiti and Tecchiolli 1994] Battiti R., Tecchiolli G.: "Simulated Annealing and Tabu Search in the Long Run: A Comparison on QAP Tasks"; Computer and Mathematics with Applications ,28, 6(1994), 1-8

[Bazaraa and Sherali 1982] Bazaraa M. S., Sherali H. D.: "On the Use of Exact and Heuristic Cutting Plane Methods for the Quadratic Assignment problem"; Journal of the Operational Research Society, 33(1982), 991-1003

[Boeringer and Werner 2004] Boeringer D. W., Werner D. H.: "Particle Swarm Optimization Versus Genetic Algorithms for Phased Array Synthesis"; IEEE Transactions on Antennas and Propagation, 52, 3(2004), 771-779

[Bos 1993] Bos J.: "A Quadratic Assignment Problem Solved by Simulated Annealing"; Journal of Environmental Management, 37, 2(1993), 127-145

[Boülte and Thonemann 1996] Boülte A., Thonemann U. W.: " Optimizing Simulated Annealing Schedules with Genetic Programming"; European Journal of Operational Research, 92, 2(1996), 402-416

[Bozer and Suk-Chul 1996] Bozer Y. A., Suk-Chul R.: "A Branch and Bound Method for Solving the Bidirectional Circular Layout Problem"; Applied Mathematical Modeling, 20, 5(1996), 342-351

[Brusco and Stahl 2000] Brusco M. J., Stahl S.: " Using Quadratic Assignment Methods to Generate Initial Permutations for Least-Squares Unidimensional Scaling of Symmetric Proximity Matrices"; Journal of Classification, 17, 2(2000), 197-223

[Burkard and Rendl 1984] Burkard R. E., Rendl F.: "A Thermodynamically Motivated Simulation Procedure for Combinatorial Optimization Problems"; European Journal of Operational Research, 17, 2(1984), 169-174

[Chakrapani and Skorin-Kapov 1993] Chakrapani J., Skorin-Kapov J.: "Massively Parallel Tabu Search for the Quadratic Assignment Problem"; Annals of Operations Research, 41, 1-4(1993), 327-342

[Ciriani et al. 2004] Ciriani V., Pisanti N., Bernasconi A.: "Room Allocation: A polynomial Subcase of the Quadratic Assignment Problem"; Discrete Applied Mathematics, 144, 3 (2004), 263-269

[Çela 1998] Çela E.: "The Quadratic Assignment Problem: Theory and Algorithms"; "Combinatorial Optimization" (Du, D.Z. Pardalos, P. Eds.), Kluwer Academic Publishers, Dordrecht (1998).

[Chakrapani and Skorin-Kapov 1993] Chakrapani J., Skorin-Kapov J.: "Massively Parallel Tabu Search for the Quadratic Assignment Problem"; Annals of Operations Research, 41, 1-4(1993), 327-342

[Clausen and Perregaard 1997] Clausen J., Perregaard M.: "Solving Large Quadratic Assignment Problems in Parallel"; Computational Optimization and Applications, 8,(1997), 111-127

[Clerc and Kennedy 2002] Clerc M., Kennedy J.: "The Particle Wwarm-explosion, Stability, and Convergence in A Multidimensional Complex Space"; IEEE Transactions on Evolutionary Computation, 6, 1(2002), 58-73

[Clerc 2006] Clerc M.: " Particle Swarm Optimization"; ISTE Publishing Company, London (2006).

[Cristian 2003] Cristian T. I.: "The Particle Swarm Optimization Algorithm: Convergence Analysis and Parameter Selection"; Information Processing Letters, 85, 6 (2003), 317-325

[Cung et al. 1997] Cung V. D., Mautor T., Michelon P., Tavares A.: "A Scatter Search Based Approach for the Quadratic Assignment Problem"; Proceedings of IEEE International Conference on Evolutionary Computation, (1997), 165-169

[Dorigo and Stützle 2004] Dorigo M., Stützle, T.: "Ant Colony Optimization"; MIT Press, MA (2004).

[Drezner et al. 2003] Drezner Z.: "A New Genetic Algorithm for the Quadratic Assignment Problem"; Informs Journal on Computing, 15, 3(2003), 320-330

[Drezner 2005] Drezner Z.: "The Extended Concentric Tabu for the Quadratic Assignment Problem"; European Journal of Operational Research, 160,(2005), 416-422

[Drezner et al. 2005a] Drezner Z.: "Compounded Genetic Algorithms for the Quadratic Assignment Problem"; Operations Research Letters, 33, 5(2005), 475-480

[Drezner et al. 2005b] Drezner Z., Hann P.M., Taillard E.D.: "Recent Advances for the Quadratic Assignment Problem with Special Emphasis on Instances That Are Difficult for Meta-heuristic Methods"; Annals of Operations Research, 139 (2005), 65-94

[Eberhart and Shi 1998] Eberhart R. C., Shi Y.: "Comparison Between Genetic Algorithms and Particle Swarm Optimization"; Proceedings of IEEE International Conference on Evolutionary Computation, (1998), 611-616

[El-Baz 2004] El-Baz M. A.: "A Genetic Algorithm for Facility Layout Problems of Different Manufacturing Environments"; Computers and Industrial Engineering 47, 2-3(2004), 233-246

[Erdoğan and Tansel 2007] Erdoğan G., Tansel B.: "A Branch-and-cut Algorithm for Quadratic Assignment Problems Based on Linearizations"; Computers and Operations Research, 34 (2007), 1085-1106

[Fedjki and Duffuaa 2004] Fedjki C. A., Duffuaa S. O.: "An Extreme Point Algorithm for A Local Minimum Solution to the Quadratic Assignment Problem"; European Journal of Operational Research, 156, 3(2004), 566-578

[Fleurent and Glover 1999] Fleurent C., Glover, F.: " Improved Constructive Multistart Strategies for the Quadratic Assignment Problem Using Adaptive Memory"; INFORMS Journal on Computing, 11(1999), 189-203

[Gambardella et al. 1999] Gambardella L. M., Taillard E. D., Dorigo, M.:" Ant Colonies for the Quadratic Assignment Problem"; Journal of the Operational Research Society, 50, (1999), 167-176

[Gilmore 1962] Gilmore, P. C.: "Optimal and Suboptimal Algorithms for the Quadratic Assignment Problem"; SIAM Journal on Applied Mathematics, 10 (1962), 305-313

[Grosan et al. 2005] Grosan C., Abraham A., Nicoara M.: "Search Optimization Using Hybrid Particle Sub-swarms and Evolutionary Algorithms"; International Journal of Simulation Systems, Science & Technology, 6, 10-11(2005),60-79

[Guo and Tang 2001] Guo C., Tang H.: " Global Convergence Properties of Evolution Stragtegies"; Mathematica Numerica Sinica, 23, 1(2001), 105-110

[Gutin and Yeo 2002] Gutin G., Yeo A.: "Polynomial Approximation Algorithms for TSP and QAP with A Factorial Domination Number"; Discrete Applied Mathematics, 119, 1-2 (2002) ,107-116

[Loiola et al. 2007] Loiola E. M., Maia de Abreu N. M., Boaventura-Netto P. O., Hahn P., Querido, T.: "A Survey for the Quadratic Assignment Problem"; European Journal of Operational Research, 176, (2007), 657-690

[Hansen and Mladenović 2001] Hansen P. , Mladenović, N.: " Variable Neighbourhood Search: Principles and Applications"; European Journal of Operations Research, 130, (2001), 449-467

[Hansen and Mladenović 2003] Hansen P., Mladenović N.: "Variable Neighbourhood Search"; In: Glover, F.W. and Kochenberger G.A. (eds.), Handbook of Metaheuristics, Dordrecht, Kluwer Academic Publishers.

[Hasegawa et al. 2002] Hasegawa M., Ikeguchi T., Aihara, K., Itoh, K.: "A Novel Chaotic Search for Quadratic Assignment Problems"; European Journal of Operational Research, 139, (2002), 543-556

[He et al. 2005] He R., Wang Y., Wang Q., Zhou J., Hu C,.: "An Improved Particle Dwarm Optimization Based on Self-adaptive Escape Velocity"; Journal of Software, 16, 12(2005), 2036-2044

[Heffley 1980] Heffley D. R.: "Decomposition of the KoopmansCBeckmann Problem"; Regional Science and Urban Economics, 10, 4(1980), 571-580.

[Jiang et al. 2005] Jiang C. W. and Etorre B.: "A Hybrid Method of Chaotic Particle Swarm Optimization and Linear Interior for Reactive Power Optimisation"; Mathematics and Computers in Simulation, 68, (2005) 57-65

[Jünger and Kaibel 2001] Jünger M., Kaibel V.: "Box-inequalities for Quadratic Assignment Polytopes"; Mathematical Programming, 91, 1(2001) 175-197

[Kelly et al. 1994] Kelly J. P., Laguna M., Glover F.: "A Study of Diversification Strategies for the Quadratic Assignment Problem"; Computers and Operations Research, 21, 8(1994) 885-893

[Kennedy and Eberhart 1997] Kennedy J., Eberhart R. C.: " A Discrete Binary Version of the Particle Swarm Algorithm"; Proceedings of International Conference on Systems, Man, and Cybernetics, IEEE Computer Society Press, (1997), 4104-4108

[Kennedy and Eberhart 2001] Kennedy J., Eberhart R.: "Swarm Intelligence"; Morgan Kaufmann, CA (2001).

[Kochhar et al. 1998] Kochhar J. S., Foster B. T., Heragu S. S.: "A Genetic Algorithm for the Unequal Area Facility Layout problem"; Computers and Operations Research, 25, 7-8(1998), 583-594

[Koopmans et al. 1957] Koopmans T. C., Beckman M. J.: " Assignment Problems and the Location of Economic Activities"; Econometrica, 25 (1957), 53-76

[Lawler 1963] Lawler E. L.: "The Quadratic Assignment Problem"; Management Science, 9, (1963), 586-599

[Liang et al. 2006] Liang J. J., Qin A. K., Suganthan P. N., Baskar S.: " Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions"; IEEE Transactions on Evolutionary Computation, 10, 3(2006), 281-295

[Lim et al. 1998] Lim M., Yu Y., Omatu S.: "Extensive Testing of a Hybrid Genetic Algorithm for Solving Quadratic Assignment Problems"; Computational Optimization and Applications, 23, (2002), 47-64

[Liu et al. 2006a] Liu H. Li B., Ji Y., Sun T.: "Particle Swarm Optimisation from Lbest to Gbest"; In: Abraham, A., Baets, B.D., Köppen, M., Nickolay, B. (Eds.), Applied Soft Computing Technologies: The Challenge of Complexity, Series: Advances in Soft Computing, Springer, (2006), 537-545

[Liu et al. 2006b] Liu H., Abraham A., Choi O., Moon S. H.: "Variable Neighborhood Particle Swarm Optimization for Multi-objective Flexible Job-shop Scheduling Problems"; Proceedings of The Sixth International Conference on Simulated Evolution and Learning, SEAL 2006, Lect. Notes in Comp. Sci, 4247(2006), 197-204

[Liu et al. in press a] Liu H., Abraham A., Clerc M.: "Chaotic Dynamic Characteristics in Swarm Intelligence"; Applied Soft Computing Journal, Available online `http://dx.doi.org/10.1016/j.asoc.2006.10.006`.

[Liu et al. in press b] Liu H., Abraham A., Zhang J.: "A Particle Swarm Approach to Quadratic Assignment Problems"; Proceedings of The Eleventh Online World Conference on Soft Computing in Industrial Applications (WSC11), Springer (in press).

[Maniezzo and Colorni 1999] Maniezzo V., Colorni A.: "The Ant System Applied to the Quadratic Assignment Problem"; IEEE Transactions on Knowlege and Data Engineering, 11, 5(1999), 769-778

[Margarita and Carlos 2006] Margarita R., Carlos A. C. C.: "Multi-objective Particle Swarm Optimizers: A Survey of the State-of-the-Art"; International Journal of Computational Intelligence Research, 2, 3(2006), 287-308

[Mautor and Roucairol 1994] Mautor T., Roucairol C.: "A New Exact Algorithm for the Solution of Quadratic Assignment Problems"; Discrete Applied Mathematics, 55, (1994), 281-293

[Merz and Freisleben 1999] Merz P., Freisleben B.: "A Comparison of Mimetic Algorithms, Tabu Search, and Ant Colonies for the Quadratic Assignment Problem"; Proceedings of the 1999 International Congress of Evolutionary Computation , IEEE Press, (1999), 2063-2070

[Merz and Freisleben 2000] Merz P., Freisleben B.: "Fitness Landscape Analysis and Mimetic Algorithms for the Quadratic Assignment Problem"; IEEE Transactions on Evolutionary Computation, 4, 4 (2000), 337-352

[Middendorf et al. 2002] Middendorf M., Reischle F., Schmeck H.: "Multi Colony Ant Algorithms"; Journal of Heuristics, 8, 3(2002), 305-320

[Mills et al. 2003] Mills P., Tsang E., Ford J., : "Applying an Extended Guided Local Search to the Quadratic Assignment Problem"; Annals of Operations Research, 118, 1-4(2003), 121-135.

[Miranda et al. 2005] Miranda G., Luna H. P. L., Mateus G. R., Ferreira R. P. M.: "A Performance Guarantee Heuristic for Electronic Components Placement Problems Including Thermal Effects"; Computers and Operations Research, 32, (2005), 2937-2957

[Misevicius 2003] Misevicius A.: "A Modified Simulated Annealing Algorithm for the Quadratic Assignment Problem"; Informatica, 14, 4 (2003), 497-514

[Misevicius 2004] Misevicius A.: "An Improved Hybrid Genetic Algorithm: New Results for the Quadratic Assignment Problem"; Knowledge-Based Systems, 17, (2004), 65-73

[Misevicius 2005] Misevicius A.: “A Tabu Search Algorithm for the Quadratic Assignment Problem”; Computational Optimization and Applications, 30, (2005), 95-111

[Nissen and Paul 1995] Nissen V., Paul H.: “A Modification of Threshold Accepting and its Application to the Quadratic Assignment Problem”; Operation Research Spektrum, 17, 2-3 (1995), 205-210

[Oliveira et al. 2004] Oliveira C. A. S., Pardalos M. P., Resende M. G. G.: “GRASP with Path Relinking for the Quadratic Assignment Problem”; Proceedings of Third International Workshop on Experimental and Efficient Algorithms, Lect. Notes Comp. Sci., 2266, (2004), 356-368

[Ostrowski and Ruoppila 1997] Ostrowski T., Ruoppila V. T.: “Genetic Annealing Search for Index Assignment in Vector Quantization”; Pattern Recognition Letters, 18, 4(1997), 311-318

[Pang et al. 2004] Pang W., Wang K., Zhou C., Dong L.: “Fuzzy Discrete Particle Swarm Optimization for Solving Traveling Salesman Problem”; Proceedings of the Fourth International Conference on Computer and Information Technology, IEEE CS Press, 796-800

[Parsopoulos and Vrahatis 2002] Parsopoulos K. E., Vrahatis M. N.: “Recent Approaches to Global Optimization Problems Through Particle Swarm Optimization”; Natural Computing, 1, (2002), 235-306

[Randall 2004] Randall M.: “ Near Parameter Free ant Colony Optimization”; Proceedings of Fourth International Workshop on Ant Colony Optimization and Swarm Intelligence, Lect. Notes Comp. Sci., 3172, (2004), 374-381

[Roucairol 1987] Roucairol C.: “A Parallel Branch and Bound Algorithm for the Quadratic Assignment Problem”; Discrete Applied Mathematics, 18, (1987), 211-225

[Salman et al. 2001] Salman A., Ahmad I., Al-Madani S.: “Particle Swarm Optimization for Task Assignment Problem”; Microprocessors and Microsystems, 26, (2002), 363-371

[Schute and Groenwold 2005] Schute J. F., Groenwold A. A.: “A Study of Global Optimization Using Particle Swarms”; Journal of Global Optimization, 31, (2005), 93-108

[Settles et al. 2003] Settles M., Rodebaugh B., Soule T.: “Comparison of Genetic Algorithm and Particle Swarm Optimizer When Evolving A Recurrent Neural Network”; Proceedings of Genetic and Evolutionary Computation Conference, GECCO 2003, Lect. Notes Comp. Sci., 2723, 148-149

[Solimanpur et al. 2004] Solimanpur M., Vrat P., Shankar R.: “Ant Colony Optimization Algorithm to the Inter-cell Layout Problem in Cellular Manufacturing”; European Journal of Operational Research, 157, 3 (2004), 592-606

[Stützle and Holger 2000] Stützle T., Holger H.: “MAX-MIN Ant System”; Future Generation Computer Systems, 16, 8 (2000), 889-914

[Stützle 2006] Stützle T.: “Iterated Local Search for the Quadratic Assignment Problem. European Journal of Operational Research, 174, (2006), 1519-1539

[Talbi et al. 2001] Talbi E. G., Rouxb O. Fonlupt, Robillard D.: “ Parallel Ant Colonies for the Quadratic Assignment Problem”; Future Generation Computer Systems, 17, (2001), 441-449

[Tate and Smith 1995] Tate D. E., Smith A. E.: “A Genetic Approach to the Quadratic Assignment Problem”; Computers and Operations Research, 22, (1995), 73-83

[Tian et al. 1996] Tian P., Wang H. C., Zhang D. M.: “Simulated Snnealing for the Quadratic Assignment Problem: A Further Study”; Computers and Industrial Engineering, 31, 3-4 (1996), 925-928

[Taillard 1991] Taillard E.: “Robust Taboo Search for the Quadratic Assignment Problem”; Parallel Computing, 17, (1991), 443-455

[Taillard 1995] Taillard E.: “Comparison of Iterative Searches for the Quadratic Assignment Problem”; Location Science, 3, 2 (1995), 87-105

[Van den Bergh and Engelbrecht 2006] Van den Bergh F., Engelbrecht A. P.: "A Study of Particle Swarm Optimization Particle Trajectories"; Information Sciences, 176, (2006), 937-971

[Weisstein 2006] Weisstein E. W.: "Borel-Cantelli Lemma"; From MathWorld – A Wolfram Web Resource, `http://mathworld.wolfram.com/Borel-CantelliLemma.html`.

[Wiesemann and Stützle 2006] Wiesemann W., Stützle T.: "Iterated Ants: An Experimental Study for the Quadratic Assignment Problem"; Proceedings of the Fifth International Workshop on Ant Colony Optimization and Swarm Intelligence, ANTS 2006, Lect. Notes Comp. Sci., 4150, (2006), 179-190

[Wikipedia 2006] Wikipedia:"NP-hard"; From Wikipedia – The Free Encyclopedia Web Resource, `http://en.wikipedia.org/wiki/NP-hard`.

[Yip and Pao 1994] Yip P. P. C., Pao Y.: "A Guided Evolutionary Simulated Annealing Approach to the Quadratic Assignment Problem"; IEEE Transaction on Systems, Man, and Cybernetics, 24, 9(1994), 1383-1387

[Yu and Sarker 2003] Yu J., Sarker B. R.: "Directional Decomposition Heuristic for a Linear Machine-cell Location Problem"; European Journal of Operational Research, 149, 1 (2003), 142-184

[Zhang et al. 2005] Zhang C., Lin Z., Lin Z.: "Variable Neighborhood Search with Permutation Distance for QAP"; Preceedings of the Ninth International Conference on Knowledge-Based Intelligent Information and Engineering Systems, KES 2005, Lect. Notes Comp. Sci., 3684, 81-88