An analysis of the inertia weight parameter for binary particle swarm optimization

Liu, Jianhua; Mei, Yi; Li, Xiaodong

Liu, Mei, Y., & Li, X. (2016). An analysis of the inertia weight parameter for binary particle swarm optimization. IEEE Transactions on Evolutionary Computation, 20(5), 666–681. https://doi.org/10.1109/TEVC.2015.2503422
Document Version: Accepted Manuscript

Published Version: https://doi.org/10.1109/TEVC.2015.2503422

An Analysis of the Inertia Weight Parameter for Binary Particle Swarm Optimization

Jianhua Liu, *Member, IEEE,* Yi Mei, *Member, IEEE* and Xiaodong Li, *Senior Member, IEEE*

Abstract—**In particle swarm optimization, the inertia weight is an important parameter for controlling its search capability. There have been intensive studies of the inertia weight in continuous optimization, but little attention has been paid to the binary case. This study comprehensively investigates the effect of the inertia weight on the performance of binary particle swarm optimization, from both theoretical and empirical perspectives. A mathematical model is proposed to analyze the behavior of binary particle swarm optimization, based on which several lemmas and theorems on the effect of the inertia weight are derived. Our research findings suggest that in the binary case, a smaller inertia weight enhances the exploration capability while a larger inertia weight encourages exploitation. Consequently, this paper proposes a new adaptive inertia weight scheme for binary particle swarm optimization. This scheme allows the search process to start first with exploration and gradually move towards exploitation by linearly increasing the inertia weight. The experimental results on 0/1 knapsack problems show that the binary particle swarm optimization with the new increasing inertia weight scheme performs significantly better than that with the conventional decreasing and constant inertia weight schemes. This study verifies the efficacy of increasing inertia weight in binary particle swarm optimization.**

Index Terms—**Binary Particle Swarm Optimization, Mathematical Modelling, Knapsack Problems.**

I. Introduction

PARTICLE Swarm Optimization (PSO) [1] [2] has been successfully applied to many optimization problems, in particular those with continuous domains. It has also been applied to discrete domains (e.g. [3] [4] [5] [6] [7] [8]), and extended from single-objective to multi-objective optimization (e.g. [7] [8] [9]). Among the various discrete PSO variants, the Binary PSO (BPSO), originally developed by Kennedy and Eberhart [5], is probably the most well-known. BPSO has been applied to many combinatorial problems, e.g. job-shop scheduling [10], the knapsack problem [11], electric power systems [12], data mining (more specifically, feature selection) [13] [14], image processing [15], sensor networks [16], very-large-scale integrated circuit design [17], and others [18] [19]. Nevertheless, the success of BPSO for solving combinatorial problems is still limited compared with Continuous PSO (CPSO) [20].

From a theoretical perspective, CPSO has been studied extensively. For example, dynamic system theory was adopted to analyze CPSO in [21]. The trajectory of a particle was analyzed in [22], which led to a model of constriction PSO. The studies on a particle's trajectory was further extended in [23] to include the influence of the inertia weight. All the above theoretical analyses were based on the stagnation assumption, in which the personal and global best positions were assumed to remain the same throughout the process. There are also other studies in which the stagnation assumption was relaxed to some extent ([24] [25] [26] [27] [28]). However, to the best of our knowledge, the theoretical analysis of BPSO is quite limited.

In PSO, the inertia weight plays an important role for balancing its exploration and exploitation. The fact was first mentioned and discussed in [29]. Typically, the balance between exploration and exploitation in CPSO is related to both the inertia weight and the acceleration coefficients [23]. However, the acceleration coefficients are often set to equal constant values, a common practice in CPSO [30] [31]. In such a case, a larger inertia weight would improve CPSO's exploration ability, while a smaller inertia weight would emphasize more on exploitation [30] [31] [32] [33]. Based on this observation, a linearly decreasing inertia weight scheme was proposed in [30], and validated for CPSO in [32] [33] [34], where the acceleration coefficients were set to equal constant values. The decreasing inertia weight with equal constant acceleration coefficients has been widely accepted as a norm by many CPSO researchers and practitioners [31] [35] [36]. Zheng et al. [37] [38] recommended an increasing inertia weight scheme. However, they used different acceleration coefficient values, which made it difficult to analyze how the inertia weight affects the performance of the algorithm [39]. Although there are other adaptive inertia weight schemes proposed (e.g. [40] [41]), the decreasing inertia weight is still one of the most widely adopted schemes [42].

With regard to BPSO, no attempts have been made to investigate the effect of the inertia weight parameter on the trade-off between exploration and exploitation of the algorithm. When designing an adaptive inertia weight scheme for BPSO, most studies directly adopted the commonly used decreasing inertia weight scheme in CPSO (e.g. benchmark functions [43], distribution network reconfiguration [44], feature selection [45], wind turbines placement [46]). However, there is no theoretical study or analysis found to support the validity of such a direct adoption.

Apart from a direct adoption of the linearly decreasing inertia weight setting from CPSO, several studies also suggest other settings. For example, a review on advances in particle swarm optimization for antenna designs suggested a constant weight of 1.0 to be used [47]. In another study [48], it was found that the inertia weight with a value less than 1.0 prevents

Jianhua Liu is with the School of Information Science and Engineering, Fujian University of Technology, Fuzhou, Fujian, 350108, China. e-mail: jhliu@fjnu.edu.cn.

Yi Mei is with the School of Engineering and Computer Science, Victoria University of Wellington, Wellington 6140, New Zealand. e-mail: yi.mei@ecs.vuw.ac.nz.

Xiaodong Li is with the School of Computer Science and Information Technology, RMIT University, Melbourne, VIC, 3001, Australia. e-mail: xiaodong.li@rmit.edu.au.

Manuscript received ** **, **; revised ** **, ****.

convergence. A chaotic BPSO was developed for feature selection in [49], which utilized chaotic maps to determine the inertia weight. Overall, with no theoretical guidance, it is difficult to design an efficient inertia weight scheme for BPSO.

To find a guideline for setting the inertia weight in BPSO, this paper carries out first-order analysis on the effect of the inertia weight on the behavior of BPSO, based on which a novel adaptive inertia weight scheme for BPSO is proposed. In short, our research findings suggest that in general a smaller inertia weight encourages exploration while a larger inertia weight enhances exploitation. Thus, we recommend that in most cases an increasing inertia weight should be favorably considered for BPSO. Our experimental results on the classical 0/1 knapsack problems are consistent with this conclusion on the inertia weight setting in BPSO. This research makes the following contributions:

1) A theoretical study of BPSO is carried out, where several lemmas and theorems are derived to provide evidence that the inertia weight significantly influences the velocity term of BPSO under different scenarios.
2) The impact of the inertia weight on the performance of BPSO is investigated. Generally speaking, with the acceleration coefficients being fixed, a smaller inertia weight tends to enhance exploration, whereas a larger inertia weight is more likely to encourage exploitation.
3) A BPSO with an increasing inertia weight is proposed, and evaluated on the 0/1 knapsack problems. The empirical results showed that our idea (increasing the inertia weight) obtained from the theoretical results is effective in practice.

The rest of the paper is organized as follows: Section II introduces the standard framework of BPSO, which is proposed by Kennedy and Eberhart [5]. Section III presents the theoretical studies on BPSO, including a set of lemmas and theorems regarding the effect of the inertia weight on the behaviors of BPSO under different scenarios. Remarks are also given to explain the implications of the lemmas and theorems. Then, Section IV proposes a BPSO with a linearly increasing inertia weight scheme, based on the conclusions of our theoretical studies. Section V carries out the experimental studies on the 0/1 knapsack problems, which demonstrates the efficacy of the proposed linearly increasing inertia weight scheme. Finally, Section VI provides the conclusions and future work.

II. The standard BPSO

The standard BPSO was originally developed by Kenndey and Eberhart [5] in 1997. In BPSO, a particle is represented as a bit string, in which each bit can take the value of either 0 or 1. Unlike in CPSO, the position of a particle in BPSO is updated by switching each bit value between 0 and 1 based on the velocity of that bit. Specifically, for the d^{th} bit of the i^{th} particle, the velocity v_{id} is transformed to a probability $s(v_{id})$ of taking the value of 1 by the following sigmoid transfer function:

$$s(v_{id}) = \frac{1}{1+e^{-v_{id}}}. \tag{1}$$

Based on $s(v_{id})$, the bit value x_{id} is updated as follows:

$$x_{id} = \begin{cases} 1, & \text{if } rand() \leq s(v_{id}), \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

where $rand()$ randomly samples a value from the uniform distribution within the interval of $[0,1]$. In other words, x_{id} takes 1 with a probability of $s(v_{id})$.

During the search process of BPSO, v_{id} is updated according to the following rule:

$$v_{id} = wv_{id} + c_1 r_{1d}(p_{id} - x_{id}) + c_2 r_{2d}(g_d - x_{id}), \tag{3}$$

where $0 < w \leq 1$ is the inertia weight. p_{id} stands for the d^{th} bit of the personal best position of the i^{th} particle. g_d denotes the d^{th} bit of the global best position (i.e. the best personal best position among that of all the particles). $c_1 > 0$ and $c_2 > 0$ are the acceleration coefficients. r_{1d} and r_{2d} are random variables which follow the uniform distribution between 0 and 1.

In practice, problem-dependent velocity clamping techniques (e.g. [50]) are often used to prevent too large velocities. In this paper, after being updated by Eq. (3), the velocity v_{id} is bounded by a threshold $\hat{v}$ as follows:

$$v_{id} = \begin{cases} \hat{v}, & \text{if } v_{id} > \hat{v}, \\ -\hat{v}, & \text{if } v_{id} < -\hat{v}. \end{cases} \tag{4}$$

In Eq. (3), the inertia weight w was originally proposed for CPSO [30] to encourage exploration at the start of the search, and then gradually move towards exploitation. To this end, a linearly decreasing inertia weight scheme was proposed [34] as follows:

$$w = \overline{w} - (\overline{w} - \underline{w})\frac{\pi}{\overline{\pi}}, \tag{5}$$

where π and $\overline{\pi}$ stand for the number of iterations elapsed so far and the maximal number of iterations respectively. $\overline{w}$ and $\underline{w}$ are the predefined upper and lower bounds of the inertia weight respectively. The above decreasing inertia weight scheme has been widely adopted in both CPSO and BPSO, to update the inertia weight before updating the velocity in Eq. (3). The pseudo code of the standard framework of BPSO is described in Algorithm 1, where $\mathbf{x}_i = (x_{i1}, \ldots, x_{in})$ (n is the dimension) and $\mathbf{p}_i = (p_{i1}, \ldots, p_{in})$ stand for the current position and personal best position of the i^{th} particle respectively. $\mathbf{g} = (g_1, \ldots, g_n)$ is the global best position.

III. Theoretical analysis of BPSO

In BPSO, a particle is represented as a bit string. Each bit is updated by Eqs. (1)–(3). To simplify analysis, we adopt the stagnation assumption, which is commonly used for analyzing PSO. That is, all the p_{id}'s and g_d's in Eq. (3) are fixed throughout the process. In this case, we can safely remove the bit index d in Eq. (3), and rewrite it as follows:

$$v_{t+1} = wv_t + c_1 r_1(p - x_t) + c_2 r_2(g - x_t), \tag{6}$$

where v_t and x_t are the velocity and the value of the considered bit in the t^{th} iteration. In standard BPSO, c_1, c_2 and w are set to constants.

In this section, we will analyse the behavior of BPSO under different w, p and g values. The velocity v_t is updated by Eq.

Algorithm 1: The standard framework of BPSO for minimization

```
Randomly generate an initial population;
Randomly generate the initial velocities within the
  velocity bound;
repeat
    for $i = 1$ to Population Size do
        if $f(\mathbf{x}_i) < f(\mathbf{p}_i)$ then $\mathbf{p}_i = \mathbf{x}_i$;
        if $f(\mathbf{p}_i) < f(\mathbf{g})$ then $\mathbf{g} = \mathbf{p}_i$;
    end
    for $i = 1$ to Population Size do
        for $d = 1$ to Dimension Size do
            Calculate $w$ using Eq. (5);
            Update velocity with Eq. (3);
            Update position using Eq. (1) and Eq. (2);
        end
    end
until termination criterion is met;
```

(6) and the distribution of the position x_t is defined as follows:

$$\Pr(x_t = 1) = \frac{1}{1+e^{-v_t}}, \tag{7}$$

$$\Pr(x_t = 0) = 1 - \frac{1}{1+e^{-v_t}}. \tag{8}$$

A. Behavior of v when $w = 1$

Lemma 1. *If $w = 1$ and $p = g = 1$, then $\forall\, t > 0$, $v_{t+1} \geq v_t$.*

Proof: From Eq. (6), when $w = 1$, and $p = g = 1$,

$$v_{t+1} = v_t + c_1 r_1 (1 - x_t) + c_2 r_2 (1 - x_t). \tag{9}$$

Since $\forall\, t > 0$, $x_t \leq 1$, and $c_1 > 0$, $c_2 > 0$, $r_1 \geq 0$, $r_2 \geq 0$, then

$$c_1 r_1 (1 - x_t) + c_2 r_2 (1 - x_t) \geq 0,$$

and thus

$$v_{t+1} \geq v_t.$$

■

Remark. *Lemma 1 indicates that when $w = 1$ and $p = g = 1$, v_t is monotonically non-decreasing. According to Eq. (7),* $\Pr(x_{t+1} = 1)$ *is no less than* $\Pr(x_t = 1)$.

Lemma 2. *If $w = 1$ and $p = g = 1$, then $\forall\, \overline{v} > v_0$, $\exists\, T > 0$, so that $\forall\, t > T$, $E[v_t] > \overline{v}$.*

Proof: If $v_{t-1} \geq \overline{v}$, then from Lemma 1, $v_t \geq v_{t-1} \geq \overline{v}$. Therefore, $E[v_t] > \overline{v}$ holds.

If $v_{t-1} < \overline{v}$, then from Lemma 1, $\forall\, i \in \{0, \ldots, t-2\}$, $v_i \leq v_{t-1} < \overline{v}$, and

$$1 - \frac{1}{1+e^{-v_i}} > 1 - \frac{1}{1+e^{-\overline{v}}}.$$

From Eq. (9), we have

$$v_t = v_0 + \sum_{i=0}^{t-1} (c_1 r_1 + c_2 r_2)(1 - x_i).$$

Therefore,

$$\begin{aligned} E[v_t] &= v_0 + \frac{c_1 + c_2}{2} \sum_{i=0}^{t-1} E[1 - x_i] \\ &= v_0 + \frac{c_1 + c_2}{2} \sum_{i=0}^{t-1} \Pr(x_i = 0) \\ &= v_0 + \frac{c_1 + c_2}{2} \sum_{i=0}^{t-1} \left(1 - \frac{1}{1+e^{-v_i}}\right). \end{aligned}$$

Since $c_1 > 0$, $c_2 > 0$,

$$\begin{aligned} E[v_t] &= v_0 + \frac{c_1 + c_2}{2} \sum_{i=0}^{t-1} \left(1 - \frac{1}{1+e^{-v_i}}\right) \\ &> v_0 + \frac{c_1 + c_2}{2} \sum_{i=0}^{t-1} \left(1 - \frac{1}{1+e^{-\overline{v}}}\right) \\ &= v_0 + \frac{c_1 + c_2}{2} \left(1 - \frac{1}{1+e^{-\overline{v}}}\right) t. \end{aligned}$$

Let

$$T = \frac{\overline{v} - v_0}{\frac{c_1 + c_2}{2}\left(1 - \frac{1}{1+e^{-\overline{v}}}\right)} > 0,$$

then $\forall\, t > T$,

$$\begin{aligned} E[v_t] &> v_0 + \frac{c_1 + c_2}{2} \left(1 - \frac{1}{1+e^{-\overline{v}}}\right) t \\ &> v_0 + \frac{c_1 + c_2}{2} \left(1 - \frac{1}{1+e^{-\overline{v}}}\right) T = \overline{v}. \end{aligned}$$

■

Remark. *Lemma 2 implies that when $w = 1$ and $p = g = 1$, v_t is expected to diverge. For any upper bound $\overline{v}$, when t is sufficiently large, v_t is expected to be larger than $\overline{v}$. According to Eq. (7),* $\Pr(x_{t+1} = 1)$ *is expected to converge to* 1.

Lemma 3. *If $w = 1$ and $p = g = 0$, then $\forall\, t > 0$, $v_{t+1} \leq v_t$.*

Proof: From Eq. (6), when $w = 1$, and $p = g = 0$,

$$v_{t+1} = v_t - c_1 r_1 x_t - c_2 r_2 x_t. \tag{10}$$

Since $\forall\, t > 0$, $x_t \geq 0$, and $c_1 > 0$, $c_2 > 0$, $r_1 \geq 0$, $r_2 \geq 0$, then

$$-c_1 r_1 x_t - c_2 r_2 x_t \leq 0,$$

and

$$v_{t+1} \leq v_t.$$

■

Remark. *Lemma 3 suggests that when $w = 1$ and $p = g = 0$, v_t is monotonically non-increasing. According to Eq. (8),* $\Pr(x_{t+1} = 0)$ *is no less than* $\Pr(x_t = 0)$.

Lemma 4. *If $w = 1$ and $p = g = 0$, then $\forall\, \underline{v} < v_0$, $\exists\, T > 0$, so that $\forall\, t > T$, $E[v_t] < \underline{v}$.*

Proof: If $v_{t-1} \leq \underline{v}$, then from Lemma 3, $v_t \leq v_{t-1} \leq \underline{v}$. Therefore, $E[v_t] < \underline{v}$ holds.

If $v_{t-1} > \underline{v}$, then from Lemma 3, $\forall\, i \in \{0, \ldots, t-2\}$, $v_i \geq v_{t-1} > \underline{v}$, and

$$\frac{1}{1+e^{-v_i}} > \frac{1}{1+e^{-\underline{v}}}.$$

From Eq. (10), we have

$$v_t = v_0 - \sum_{i=0}^{t-1}(c_1 r_1 + c_2 r_2)x_i.$$

Therefore,

$$\begin{aligned} E[v_t] &= v_0 - \frac{c_1+c_2}{2}\sum_{i=0}^{t-1}E[x_i] \\ &= v_0 - \frac{c_1+c_2}{2}\sum_{i=0}^{t-1}\Pr(x_i=1) \\ &= v_0 - \frac{c_1+c_2}{2}\sum_{i=0}^{t-1}\left(\frac{1}{1+e^{-v_i}}\right). \end{aligned}$$

Since $c_1 > 0$, $c_2 > 0$,

$$\begin{aligned} E[v_t] &= v_0 - \frac{c_1+c_2}{2}\sum_{i=0}^{t-1}\left(\frac{1}{1+e^{-v_i}}\right) \\ &< v_0 - \frac{c_1+c_2}{2}\sum_{i=0}^{t-1}\left(\frac{1}{1+e^{-\underline{v}}}\right) \\ &= v_0 - \frac{c_1+c_2}{2}\left(\frac{1}{1+e^{-\underline{v}}}\right)t. \end{aligned}$$

Let

$$T = \frac{v_0 - \underline{v}}{\frac{c_1+c_2}{2}\left(\frac{1}{1+e^{-\underline{v}}}\right)} > 0,$$

then $\forall\, t > T$,

$$\begin{aligned} E[v_t] &< v_0 - \frac{c_1+c_2}{2}\left(\frac{1}{1+e^{-\underline{v}}}\right)t \\ &< v_0 - \frac{c_1+c_2}{2}\left(\frac{1}{1+e^{-\underline{v}}}\right)T = \underline{v}. \end{aligned}$$

■

Remark. *Lemma 4 implies that when $w = 1$ and $p = g = 0$, v_t is expected to diverge. For any lower bound $\underline{v}$, when t is sufficiently large, v_t is expected to be smaller than $\underline{v}$. According to Eq. (8),* $\Pr(x_{t+1} = 0)$ *is expected to converge to* 1.

Lemma 5. *If $w = 1$ and $p \neq g$, then $\forall\, t_0$, $t > 0$,*

$$v_{t_0} + \beta_1 t \leq E[v_{t_0+t}] \leq v_{t_0} + \beta_2 t,$$

where

$$\begin{aligned} \beta_1 &= \frac{1}{2}\left(\min(c_1,c_2) - \frac{c_1+c_2}{1+e^{-v_{max}}}\right), \\ \beta_2 &= \frac{1}{2}\left(\max(c_1,c_2) - \frac{c_1+c_2}{1+e^{-v_{min}}}\right), \\ v_{max} &= \max_{i\in\{t_0,\ldots,t_0+t-1\}} v_i, \\ v_{min} &= \min_{i\in\{t_0,\ldots,t_0+t-1\}} v_i. \end{aligned}$$

Proof: First, we assume that $p = 1$ and $g = 0$. Then

$$v_{t+1} = v_t + c_1 r_1 - (c_1 r_1 + c_2 r_2)x_t,$$

and $\forall\, t_0$, $t > 0$,

$$v_{t_0+t} = v_{t_0} + \sum_{i=t_0}^{t_0+t-1}(c_1 r_1 - (c_1 r_1 + c_2 r_2)x_i).$$

Then,

$$\begin{aligned} E[v_{t_0+t}] &= v_{t_0} + \sum_{i=t_0}^{t_0+t-1}\left(\frac{c_1}{2} - \frac{c_1+c_2}{2}E[x_i]\right) \\ &= v_{t_0} + \frac{1}{2}\sum_{i=t_0}^{t_0+t-1}(c_1 - (c_1+c_2)\Pr(x_i=1)) \\ &= v_{t_0} + \frac{1}{2}\sum_{i=t_0}^{t_0+t-1}\left(c_1 - \frac{c_1+c_2}{1+e^{-v_i}}\right). \end{aligned}$$

Since $\forall\, i \in \{t_0,\ldots,t_0+t-1\}$, $v_{min} \leq v_i \leq v_{max}$, then

$$\frac{1}{1+e^{-v_{min}}} \leq \frac{1}{1+e^{-v_i}} \leq \frac{1}{1+e^{-v_{max}}}.$$

And $c_1 > 0$, $c_2 > 0$, then

$$\begin{aligned} v_{t_0} + \frac{t}{2}\left(c_1 - \frac{c_1+c_2}{1+e^{-v_{max}}}\right) &\leq E[v_{t_0+t}] \\ &\leq v_{t_0} + \frac{t}{2}\left(c_1 - \frac{c_1+c_2}{1+e^{-v_{min}}}\right). \end{aligned}$$

Similarly, if $p = 0$ and $g = 1$,

$$\begin{aligned} v_{t_0} + \frac{t}{2}\left(c_2 - \frac{c_1+c_2}{1+e^{-v_{max}}}\right) &\leq E[v_{t_0+t}] \\ &\leq v_{t_0} + \frac{t}{2}\left(c_2 - \frac{c_1+c_2}{1+e^{-v_{min}}}\right). \end{aligned}$$

Therefore,

$$\begin{aligned} E[v_{t_0+t}] &\leq v_{t_0} + \frac{t}{2}\left(\max(c_1,c_2) - \frac{c_1+c_2}{1+e^{-v_{min}}}\right), \\ E[v_{t_0+t}] &\geq v_{t_0} + \frac{t}{2}\left(\min(c_1,c_2) - \frac{c_1+c_2}{1+e^{-v_{max}}}\right). \end{aligned}$$

That is,

$$v_{t_0} + \beta_1 t \leq E[v_{t_0+t}] \leq v_{t_0} + \beta_2 t.$$

■

Remark. *Lemma 5 implies that when $w = 1$ and $p \neq q$, the expectation of v_{t_0+t} is bounded by the range of $[v_{t_0} + \beta_1 t, v_{t_0} + \beta_2 t]$, which is determined by its preceding sequence $(v_{t_0},\ldots,v_{t_0+t-1})$. Additionally, note that β_1 and β_2 are decreasing functions of v_{max} and v_{min} respectively. This suggests the following properties:*

1) *Given the same t, if the upper bound of the sequence $(v_{t_0},\ldots,v_{t_0+t-1})$ is smaller, then the lower bound of $E[v_{t_0+t}]$ is larger;*
2) *Given the same t, if the lower bound of the sequence $(v_{t_0},\ldots,v_{t_0+t-1})$ is larger, then the upper bound of $E[v_{t_0+t}]$ is smaller;*
3) *If $v_{max} \leq -\ln\left(\frac{c_1+c_2}{\min(c_1,c_2)} - 1\right) \leq 0$, then $\beta_1 > 0$. As t increases, the lower bound of $E[v_{t_0+t}]$ becomes larger. That is, if the sequence $(v_{t_0},\ldots,v_{t_0+t-1})$ is consistently below a non-positive value $-\ln\left(\frac{c_1+c_2}{\min(c_1,c_2)} - 1\right) \leq 0$, a longer sequence leads to a larger lower bound of $E[v_{t_0+t}]$;*
4) *If $v_{min} \geq -\ln\left(\frac{c_1+c_2}{\max(c_1,c_2)} - 1\right) \geq 0$, then $\beta_2 < 0$. As t increases, the upper bound of $E[v_{t_0+t}]$ becomes smaller.*

That is, if the sequence $(v_{t_0},\ldots,v_{t_0+t-1})$ is consistently above a non-negative value $-\ln\left(\frac{c_1+c_2}{\max(c_1,c_2)}-1\right)\geq 0$, a longer sequence leads to a smaller upper bound of $E[v_{t_0+t}]$;

5) *According to properties 3) and 4), starting from $v_0=0$, the sequence $(v_0,v_1,\ldots)$ is expected to fluctuate around the area $\left[-\ln\left(\frac{c_1+c_2}{\min(c_1,c_2)}-1\right),-\ln\left(\frac{c_1+c_2}{\max(c_1,c_2)}-1\right)\right]$. When $c_1=c_2$, which is common in PSO, the sequence $(v_0,v_1,\ldots)$ is expected to fluctuate around the point 0. According to Eqs. (7) and (8), the sequences of $\Pr(x_t=1)$ and $\Pr(x_t=0)$ ($t\in\mathbb{N}$) fluctuate around 0.5.*

The implications of Lemmas 1–5 can be summarized as a theorem about $\Pr(x_t=1)$ and $\Pr(x_t=0)$ under different conditions when $w=1$. It is stated as follows:

Theorem 1. *When $w=1$,*

1) *if $p=g$, then $\Pr(x_t=p)$ ($\Pr(x_t=g)$) is a non-decreasing function of the generation t, and its expectation converges to 1;*
2) *if $p\neq g$, then under the standard parameter settings where $c_1=c_2$ and $v_0=0$, the sequences of $\Pr(x_t=1)$ and $\Pr(x_t=0)$ fluctuate around 0.5.*

Proof: The proof can be directly derived from the remarks of Lemmas 1–5. ■

B. Behavior of v when $0<w<1$

Lemma 6. *If $0<w<1$, $p=g=1$, then $\forall\, t>0$,*

$$w^t v_0 \leq v_t \leq w^t v_0 + (c_1+c_2)\frac{1-w^t}{1-w}.$$

Proof: From Eq. (6), when $p=g=1$,

$$v_t = w^t v_0 + \sum_{i=0}^{t-1} w^i (c_1 r_1 + c_2 r_2)(1-x_{t-i-1}).$$

Since $\forall\, i\in\{0,\ldots,t-1\}$, $0\leq x_{t-i-1}\leq 1$ and $0\leq r_1,r_2\leq 1$, $c_1>0$, $c_2>0$, then

$$\begin{aligned} w^t v_0 \leq v_t &\leq w^t v_0 + \sum_{i=0}^{t-1} w^i (c_1 r_1 + c_2 r_2)\\ &\leq w^t v_0 + (c_1+c_2)\sum_{i=0}^{t-1} w^i\\ &= w^t v_0 + (c_1+c_2)\frac{1-w^t}{1-w}. \end{aligned}$$

■

Lemma 7. *If $0<w<1$, $p=g=0$, then $\forall\, t>0$,*

$$w^t v_0 - (c_1+c_2)\frac{1-w^t}{1-w} \leq v_t \leq w^t v_0.$$

Proof: From Eq. (6), if $p=q=0$,

$$v_t = w^t v_0 - \sum_{i=0}^{t-1} w^i (c_1 r_1 + c_2 r_2) x_{t-i-1}.$$

Since $\forall\, i\in\{0,\ldots,t-1\}$, $0\leq x_{t-i-1}\leq 1$ and $0\leq r_1,r_2\leq 1$, $c_1>0$, $c_2>0$, then

$$\begin{aligned} w^t v_0 \geq v_t &\geq w^t v_0 - \sum_{i=0}^{t-1} w^i (c_1 r_1 + c_2 r_2)\\ &\geq w^t v_0 - (c_1+c_2)\sum_{i=0}^{t-1} w^i\\ &= w^t v_0 - (c_1+c_2)\frac{1-w^t}{1-w}. \end{aligned}$$

■

Lemma 8. *If $0<w<1$, $p\neq g$, then $\forall\, t>0$,*

$$w^t v_0 - \min(c_1,c_2)\frac{1-w^t}{1-w} \leq v_t \leq w^t v_0 + \max(c_1,c_2)\frac{1-w^t}{1-w}.$$

Proof: If $p=1$ and $q=0$, from Eq. (6),

$$v_t = w^t v_0 - \sum_{i=0}^{t-1} w^i (c_1 r_1 - (c_1 r_1 + c_2 r_2) x_{t-i-1}).$$

Since $\forall\, i\in\{0,\ldots,t-1\}$, $0\leq x_{t-i-1}\leq 1$ and $0\leq r_1,r_2\leq 1$, $c_1>0$, $c_2>0$, then $-c_2\leq c_1 r_1-(c_1 r_1+c_2 r_2)x_i\leq c_1$. Thus,

$$w^t v_0 - c_2\frac{1-w^t}{1-w} \leq v_t \leq w^t v_0 + c_1\frac{1-w^t}{1-w}.$$

Similarly, if $p=0$ and $q=1$,

$$w^t v_0 - c_1\frac{1-w^t}{1-w} \leq v_t \leq w^t v_0 + c_2\frac{1-w^t}{1-w}.$$

Therefore,

$$w^t v_0 - \min(c_1,c_2)\frac{1-w^t}{1-w} \leq v_t \leq w^t v_0 + \max(c_1,c_2)\frac{1-w^t}{1-w}.$$

■

Remark. *Lemmas 6, 7 and 8 give the range of v_t under different p and g values when $0<w<1$. Specifically, if $p=g=1$, then v_t is no smaller than $w^t v_0$. If $p=g=0$, then v_t is no larger than $w^t v_0$. If $p\neq g$, then the sequence $(v_0,v_1,\ldots)$ fluctuates around $w^t v_0$. Without any other information, it is reasonable to initialize v_0 to 0. In this case, if $p=g=1$, then $\forall\, t>0$, v_t is non-negative, and $\Pr(x_{t+1}=1)\geq 0.5$. If $p=g=0$, then $\forall\, t>0$, v_t is non-positive and $\Pr(x_{t+1}=0)\geq 0.5$. Otherwise, the sequence $(v_0,v_1,\ldots)$ fluctuates around 0, and the sequence of $\Pr(x_t=1)$ ($t\in\mathbb{N}$) is expected to fluctuate around 0.5.*

In addition, the width of the range of v_t is $(c_1+c_2)\frac{1-w^t}{1-w}$, regardless of the p and g values. Given the same t, a smaller w leads to a narrower range of v_t which is closer to 0.

Lemma 9. *If $0<w<1$, then $E[v_{t+1}-v_t]$ is a decreasing function of v_t.*

Proof: From Eq. (6), we have

$$E[v_{t+1}] = wv_t + \frac{c_1 p + c_2 g}{2} - \frac{c_1+c_2}{2}\left(\frac{1}{1+e^{-v_t}}\right),$$

$$E[v_{t+1}-v_t] = (w-1)v_t + \frac{c_1 p + c_2 g}{2} - \frac{c_1+c_2}{2}\left(\frac{1}{1+e^{-v_t}}\right). \quad (11)$$

Let $f(v_t) = E[v_{t+1} - v_t]$, then

$$f'(v_t) = (w-1) - \frac{c_1+c_2}{2}\left(\frac{1}{1+e^{-v_t}}\right)' \\ = (w-1) - \frac{c_1+c_2}{2} \cdot \frac{e^{-v_t}}{(1+e^{-v_t})^2}.$$

Since $w < 1$, $c_1 > 0$, $c_2 > 0$, and $\frac{e^{-v_t}}{(1+e^{-v_t})^2} > 0$, we have $f'(v_t) < 0$. Thus, $E[v_{t+1} - v_t]$ is a decreasing function of v_t. ■

Remark. *Fig. 1 illustrates an example of the relationship between v_t and $E[v_{t+1} - v_t]$ under different w values, when $c_1 = c_2 = 2$ and $p \neq g$. The figure clearly shows that $E[v_{t+1} - v_t]$ is a decreasing function of v_t, and a larger w leads to a flatter slope due to the larger $f'(v_t)$ value (closer to zero). Another interesting observation is that $E[v_{t+1} - v_t] = 0$ when $v_t = 0$ regardless of w. This phenomenon will be proven in Lemma 10.*

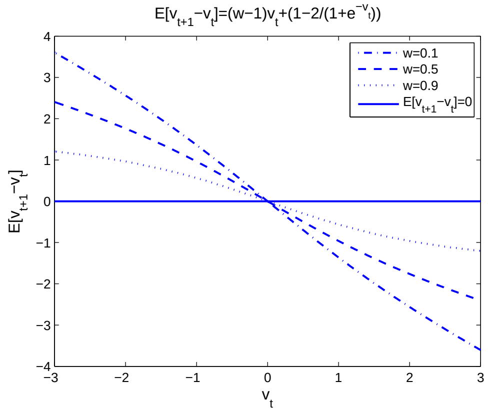


Fig. 1. The relationship between $E[v_{t+1} - v_t]$ and v_t under $w = 0.9$, 0.5 and 0.1, when $p \neq g$ and $c_1 = c_2 = 2$.

Lemma 10. *If $0 < w < 1$, then there exists a finite value v^*, so that*

$$E[v_{t+1} - v_t] \begin{cases} > 0, & \textit{if } v_t < v^*, \\ = 0, & \textit{if } v_t = v^*, \\ < 0, & \textit{if } v_t > v^*, \end{cases}$$

and v^* satisfies

$$(w-1)v^* + \frac{c_1 p + c_2 g}{2} - \frac{c_1+c_2}{2}\left(\frac{1}{1+e^{-v^*}}\right) = 0. \quad (12)$$

Proof: First, let

$$f(v) = (w-1)v,$$
$$g(v) = \frac{c_1+c_2}{2}\left(\frac{1}{1+e^{-v}}\right) - \frac{c_1 p + c_2 g}{2}.$$

Since $w < 1$, $f(v)$ is a decreasing function of v. On the other hand, it is obvious that $g(v)$ is an increasing function of v.

Then, it is easily obtained that

$$\lim_{v \to -\infty} f(v) \to \infty, \quad \lim_{v \to -\infty} g(v) = -\frac{c_1 p + c_2 g}{2},$$
$$\lim_{v \to \infty} f(v) \to -\infty, \quad \lim_{v \to \infty} g(v) = \frac{c_1+c_2}{2} - \frac{c_1 p + c_2 g}{2}.$$

Therefore, $\lim_{v \to -\infty} f(v) > \lim_{v \to -\infty} g(v)$ and $\lim_{v \to \infty} f(v) < \lim_{v \to \infty} g(v)$. Note that $f(v)$ is a decreasing function of v and $g(v)$ is an increasing function of v, there must exist a finite intersection $-\infty < v^* < \infty$, so that $f(v^*) = g(v^*)$. That is,

$$(w-1)v^* + \frac{c_1 p + c_2 g}{2} - \frac{c_1+c_2}{2}\left(\frac{1}{1+e^{-v^*}}\right) = 0.$$

The existence of a finite v^* has been proven.

According to Eqs. (11) and (12), we have

$$E[v_{t+1} - v_t | v_t = v^*] = 0.$$

Since $E[v_{t+1} - v_t]$ is a decreasing function of v_t (proven in Lemma 9), we have

$$E[v_{t+1} - v_t] \begin{cases} > 0, & \text{if } v_t < v^*, \\ = 0, & \text{if } v_t = v^*, \\ < 0, & \text{if } v_t > v^*. \end{cases}$$

■

Remark. *In the example shown in Fig. 1, since $c_1 = c_2 = 2$, one can assume $p = 1$ and $g = 0$ without loss of generality. Then,*

$$E[v_{t+1} - v_t] = (w-1)v_t + 1 - 2\left(\frac{1}{1+e^{-v_t}}\right).$$

Then, when $v_t = 0$, $E[v_{t+1} - v_t] = 0$ regardless of w. In other words, when $c_1 = c_2$ and $p \neq g$, $v^* = 0$ for all possible w values.

Lemma 10 implies that for any iteration t, if $v_t < v^*$, then the velocity is expected to increase in the next iteration. If $v_t > v^*$, the velocity is expected to decrease in the next iteration. If $v_t = v^*$, the velocity is expected to remain the same in the next iteration. Simply put, the sequence of $E[v_t]$ fluctuates around v^*.

Figs. 2 to 4 show the sequences of the mean velocity v_t ($t = 1, 2, \ldots$) over 3000 independent runs (to approximate $E[v_t]$) under different p and g values, with $w = 1$, 0.9, 0.5 and 0.1. The other parameters are set to $v_0 = 0$ and $c_1 = c_2 = 2$. The figures clearly demonstrate that when $w < 1$, the mean velocities fluctuate around the corresponding v^*'s (e.g. around 2.3 for $w = 0.9$ and $p = g = 1$, −2.3 for $w = 0.9$ and $p = g = 0$, and 0 when $p \neq g$).

The implications of Lemmas 6–10 can be summarized as a theorem about $\Pr(x_t = 1)$ and $\Pr(x_t = 0)$ under different conditions when $0 < w < 1$. It is stated as follows:

Theorem 2. *When $0 < w < 1$, then under the standard parameter settings where $c_1 = c_2$ and $v_0 = 0$,*

1) *if $p = g$, then $\Pr(x_t = p)$ ($\Pr(x_t = g)$) is always greater than 0.5;*
2) *if $p \neq g$, then the sequences of $\Pr(x_t = 1)$ and $\Pr(x_t = 0)$ fluctuate around 0.5.*

Proof: The proof can be directly derived from the remarks of Lemmas 6–10. ■

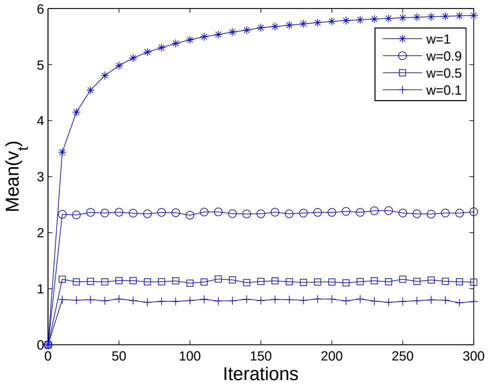


Fig. 2. The curves of the mean velocity v_t over 3000 independent runs with $w = 1$, 0.9, 0.5 and 0.1, when $v_0 = 0$, $p = g = 1$ and $c_1 = c_2 = 2$.

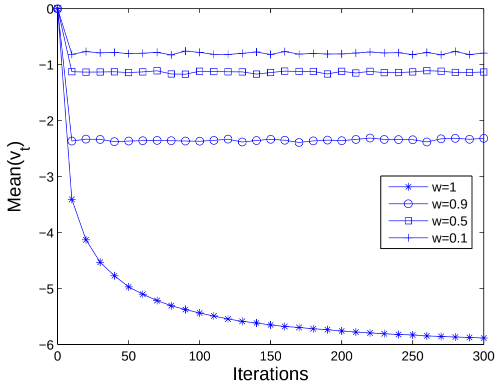


Fig. 3. The curves of the mean velocity v_t over 3000 independent runs with $w = 1$, 0.9, 0.5 and 0.1, when $v_0 = 0$, $p = g = 0$ and $c_1 = c_2 = 2$.

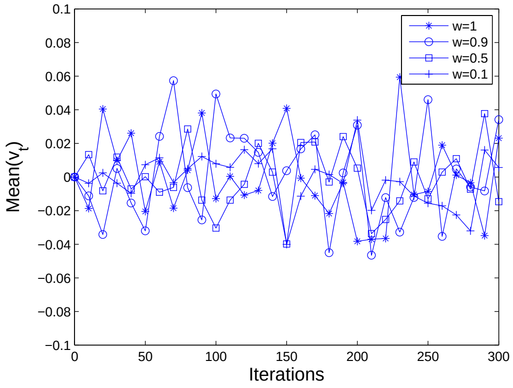


Fig. 4. The curves of the mean velocity v_t over 3000 independent runs with $w = 1$, 0.9, 0.5 and 0.1, when $v_0 = 0$, $p \neq g$ and $c_1 = c_2 = 2$.

IV. BPSO WITH AN INCREASING INERTIA WEIGHT

Section III analyzed the effect of the inertia weight w on the behavior of the velocity v and position x in the single-dimensional case (i.e. a bit), when the acceleration coefficients c_1 and c_2 are set to constants and p and g are unchanged over time. Since the velocity and position updating are carried out independently in BPSO, the lemmas and theorems derived in Section III can be directly extended to a bit string. Specifically, given an n-dimensional bit string, let $\mathbf{x} = (x_1, \ldots, x_n)$ be the position vector, $\mathbf{v} = (v_1, \ldots, v_n)$ be the velocity vector, and $\mathbf{p} = (p_1, \ldots, p_n)$ and $\mathbf{g} = (g_1, \ldots, g_n)$ be the personal and global best positions, then $\forall\, d \in \{1, \ldots, n\}$, all the lemmas and theorems in Section III can be applied to x_d, v_d, p_d and g_d. In summary, for each dimension d, we have the following properties based on the analysis in Section III:

1) If $p_d = g_d$, then a larger w is expected to lead to a larger absolute velocity (shown in Figs. 2 and 3), and thus a faster convergence speed of x_d to p_d and g_d.
2) If $p_d \neq g_d$, under the standard setting of $c_1 = c_2$ and $v_0 = 0$, $\Pr(x_d = 1)$ and $\Pr(x_d = 0)$ fluctuate around 0.5.

For the dimensions where $\mathbf{p}$ and $\mathbf{g}$ have the same values, a larger w tends to drive the corresponding x_d values towards p_d and g_d quickly. For the dimensions where $\mathbf{p}$ and $\mathbf{g}$ have different values, x_d randomly fluctuates between 0 and 1 regardless of w. However, as search continues, more and more bit values in $\mathbf{p}$ and $\mathbf{g}$ tend to converge to the same values. Thus, in the long term, given fixed c_1 and c_2 values, the convergence speed of the position $\mathbf{x}$ to the personal best position $\mathbf{p}$ of this particle and the global best position $\mathbf{g}$ tends to increase as w increases.

In a general framework of BPSO, the search starts first with exploration, and gradually moves towards exploitation as the search continues. Assuming that c_1 and c_2 are fixed, and the exploitation capability of BPSO is strengthened as w increases, an increasing inertia weight scheme is proposed to shift the search process from exploration to exploitation in BPSO.

Here, a linearly increasing scheme is adopted. In each iteration, the inertia weight w is calculated as follows:

$$w = \begin{cases} \underline{w} + \frac{\pi\cdot(\overline{w}-\underline{w})}{\rho\cdot\overline{\pi}}, & \text{if } \pi \leq \rho\cdot\overline{\pi}, \\ \overline{w}, & \text{if } \rho\cdot\overline{\pi} < \pi \leq \overline{\pi}, \end{cases} \tag{13}$$

where π and $\overline{\pi}$ stand for the number of iterations elapsed and the maximal number of iterations respectively. $\underline{w}$ and $\overline{w}$ are the lower and upper bounds of w. $0 \leq \rho \leq 1$ is the parameter to control the number of iterations to make w increase from $\underline{w}$ to $\overline{w}$. If $\rho = 0$, then w is fixed to $\overline{w}$ and there is no adaptation during the process. On the other hand, if $\rho = 1$, w linearly increases throughout the search process. Here, we set $\rho = 0.9$, which is a reasonable value to achieve a good tradeoff between exploration and exploitation according to preliminary studies.

The framework of the BPSO with the above increasing inertia weight scheme is the same as Algorithm 1. The only difference is that Eq. (5) is replaced by Eq. (13).

In addition, to make a fair comparison with the decreasing inertia weight scheme, Eq. (5) is revised to the following equation:

$$w = \begin{cases} \overline{w} - \frac{\pi\cdot(\overline{w}-\underline{w})}{\rho\cdot\overline{\pi}}, & \text{if } \pi \leq \rho\cdot\overline{\pi}, \\ \underline{w}, & \text{if } \rho\cdot\overline{\pi} < \pi \leq \overline{\pi}, \end{cases} \tag{14}$$

where ρ is set to the same value as that in Eq. (13), which is 0.9.

V. Experimental Studies

In this section, the proposed increasing inertia weight scheme is compared with the decreasing and constant inertia weight schemes on the 0/1 knapsack problems, which are commonly used benchmark problems for testing binary optimization algorithms. For the sake of convenience, the BPSOs with the three compared schemes are denoted as "Up", "Down" and "Con" respectively. Their details are given below:

- In "Up", the inertia weight is defined by Eq. (13);
- In "Down", the inertia weight is defined by Eq. (14);
- In "Con", the inertia weight is set to a proper constant value w^*, which will be determined in Section V-B by testing a range of values and selecting the best one.

A. Experimental Settings

The single- and multi-dimensional 0/1 knapsack problems are selected to test our algorithms. The Single 0/1 Knapsack Problem (SKP) can be described as follows: Given n items, each of which having a positive profit $p_i > 0$ and a positive resource consumption $r_i > 0$ (e.g. size), and a knapsack with a capacity of C, the aim is to pick a subset of items to put in the knapsack, so that the total profit of the picked items is maximized, while the total resource consumption does not exceed the capacity C of the knapsack. The problem can be formulated as follows:

$$\max \sum_{i=1}^{n} p_i x_i, \tag{15}$$

$$s.t.: \sum_{i=1}^{n} r_i x_i \leq C, \tag{16}$$

$$x_i \in \{0,1\},\ \forall\, i \in \{1,\ldots,n\}, \tag{17}$$

where the decision variable x_i takes the value of 1 if item i is picked, and 0 otherwise.

In the Multi-dimensional 0/1 Knapsack Problem (MKP), there are m ($m > 1$) resources to be considered. Each item i has a set of resource consumptions r_{ij} ($j \in \{1,\ldots,m\}$), and the knapsack has a capacity C_j ($j \in \{1,\ldots,m\}$) for each resource. Then, MKP requires that the total consumption of each resource does not exceed the corresponding capacity. The problem can be formulated as follows:

$$\max \sum_{i=1}^{n} p_i x_i, \tag{18}$$

$$s.t.: \sum_{i=1}^{n} r_{ij} x_i \leq C_j,\ \forall\, j \in \{1,\ldots,m\}, \tag{19}$$

$$x_i \in \{0,1\},\ \forall\, i \in \{1,\ldots,n\}. \tag{20}$$

MKP has a number of applications in the real world, such as the capital budgeting problem, allocation of processors and databases in a distributed computer system, project selection, cargo loading and cutting stock problems [51].

In our experiments, the SKP test dataset is obtained from an academic web page of Michigan Technological University[1], which contains 25 randomly generated instances. For MKP, the Sento, Weing and Weish test sets are selected from the MKP website of University of Nottingham[2], which include 2, 8 and 30 instances respectively. These datasets have been commonly used as the test problems of knapsack problems in previous works (e.g. [11] [52] [53] [54]).

Note that both SKP and MKP are constrained binary optimization problems. To tackle the constraints in BPSO, the penalty function strategy [55] is employed to transform the problem into an unconstrained one. Concretely, the fitness function is defined as

$$f(\mathbf{x}) = \sum_{i=1}^{n} p_i x_i + \beta \cdot \sum_{i=1}^{n} \sum_{j=1}^{m} \min(C_j - r_{ij} x_i, 0), \tag{21}$$

where β is the penalty coefficient to control the tradeoff between the objective value and the violation to the constraints. In our experimental studies, β is simply set to 10^{100}, which is a sufficiently large number to eliminate the infeasible solutions.

Tabel I describes the parameter settings in the experiments in detail. Note that the number of particles and maximal number of iterations of the BPSO are set differently for SKP and MKP instances. This is because the MKP instances have larger problem sizes, and thus require more particles and number of iterations to reach competitive results.

B. Selecting the Best Constant Inertia Weight

In order to select the best constant inertia weight for "Con", the values from $\{0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ were tested and compared. The range is selected based on that adopted by the decreasing inertia weight scheme [34], which is between 0.4 and 0.9. For each tested value, 100 independent runs of the corresponding "Con" BPSO were conducted on the Sento, Weing, and Weish MKP instances. Tables II and III show the mean value of the results obtained by "Con" with the tested inertia weights. The key features of the instances are given as well. Specifically, "m" and "n" stand for the number of resources and items in the MKP instance, and "Opt" is the optimal result given by the website[2]. In addition, the best inertia weight (with the highest mean) was compared with each of the other weights by the t-test with significance level of 0.05 and Bonferroni correction. That is, the significance level is corrected by the number of comparisons as $\alpha' = \alpha/m$, where m is the number of comparisons. For each instance, if a constant inertia weight performed significantly better than all the others, the corresponding result is marked in bold.

From the tables, one can see that 0.9 and 1.0 are the two best constant inertia weights in terms of the number of significantly better results (14 for 0.9 and 15 for 1.0). In addition, it is obvious that the best constant inertia weight depends on the problem size n. When $n \leq 60$, 0.9 is the best value in the most cases. When $n \geq 70$, however, 1.0 becomes the best value.

In addition to the number of significant better results, we also calculated the average percentage of deviation $\Delta\%$ for each instance, which is defined as follows:

$$\Delta\% = \frac{f^* - f_{mean}}{f^*}, \tag{22}$$

[1] http://www.math.mtu.edu/~kreher/cages/Data.html

[2] http://www.cs.nott.ac.uk/~jqd/mkp/index.html

TABLE I
PARAMETER SETTINGS OF THE EXPERIMENTS ON THE KNAPSACK PROBLEMS.

Parameter	**Description**	**Value**
$\overline{w}$	Upper bound of w	1
$\underline{w}$	Lower bound of w	0.4
ρ	Fraction of iterations for changing w in Eqs. (13) and (14)	0.9
β	Penalty coefficient in Eq. (21)	10^{100}
N	Number of particles in the BPSO	20 for SKP instances n for MKP instances
$\overline{\pi}$	Maximal number of iterations	1000 for SKP instances 3000 for MKP instances

TABLE II
MEAN OF THE RESULTS OF "CON" WITH DIFFERENT INERTIA WEIGHTS OVER 100 INDEPENDENT RUNS ON THE SENTO AND WEING MKP INSTANCES.

Name	m	n	Opt	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Sento1	30	60	7.7720E+03	6.3222E+03	6.6133E+03	6.8795E+03	7.2185E+03	7.4875E+03	**7.7335E+03**	7.6733E+03
Sento2	30	60	8.7220E+03	8.3238E+03	8.3634E+03	8.4246E+03	8.4917E+03	8.5862E+03	**8.6948E+03**	8.6674E+03
Weing1	2	28	1.4128E+05	1.4091E+05	1.4108E+05	1.4120E+05	1.4127E+05	1.4128E+05	1.4127E+05	1.4022E+05
Weing2	2	28	1.3088E+05	1.3027E+05	1.3066E+05	1.3082E+05	1.3088E+05	1.3088E+05	1.3088E+05	1.2771E+05
Weing3	2	28	9.5677E+04	9.4232E+04	9.4697E+04	9.5187E+04	9.5421E+04	9.5442E+04	9.5222E+04	9.0235E+04
Weing4	2	28	1.1934E+05	1.1873E+05	1.1908E+05	1.1921E+05	1.1926E+05	1.1921E+05	1.1923E+05	1.1659E+05
Weing5	2	28	9.8796E+04	9.7227E+04	9.8141E+04	9.8501E+04	9.8651E+04	9.8766E+04	9.8577E+04	9.4803E+04
Weing6	2	28	1.3062E+05	1.2999E+05	1.3025E+05	1.3038E+05	1.3048E+05	1.3051E+05	1.3051E+05	1.2761E+05
Weing7	2	105	1.0954E+06	1.0033E+06	1.0128E+06	1.0255E+06	1.0402E+06	1.0578E+06	1.0832E+06	**1.0937E+06**
Weing8	2	105	6.2432E+05	1.4546E+05	2.5105E+05	3.3927E+05	4.1678E+05	4.8488E+05	5.7694E+05	**6.0828E+05**

Under the t-test with significance level of 0.05 and Bonferroni correction, if the mean of one result is significantly better than all the others, then its corresponding mean value is highlighted in bold.

TABLE III
MEAN OF THE RESULTS OF "CON" WITH DIFFERENT INERTIA WEIGHTS OVER 100 INDEPENDENT RUNS ON THE WEISH MKP INSTANCES.

Name	m	n	Opt	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Weish01	5	30	4.5540E+03	4.4779E+03	4.5025E+03	4.5294E+03	4.5499E+03	4.5533E+03	4.5510E+03	4.5035E+03
Weish02	5	30	4.5360E+03	4.4640E+03	4.4871E+03	4.5152E+03	4.5301E+03	4.5349E+03	4.5349E+03	4.4912E+03
Weish03	5	30	4.1150E+03	4.0219E+03	4.0458E+03	4.0838E+03	4.1043E+03	4.1093E+03	4.1093E+03	4.0462E+03
Weish04	5	30	4.5610E+03	4.4495E+03	4.5012E+03	4.5380E+03	4.5607E+03	4.5610E+03	4.5610E+03	4.5397E+03
Weish05	5	30	4.5140E+03	4.3795E+03	4.4278E+03	4.4850E+03	4.5108E+03	**4.5140E+03**	**4.5140E+03**	4.4582E+03
Weish06	5	40	5.5570E+03	5.3587E+03	5.4031E+03	5.4456E+03	5.4935E+03	5.5381E+03	**5.5498E+03**	5.5031E+03
Weish07	5	40	5.5670E+03	5.3348E+03	5.3888E+03	5.4389E+03	5.4976E+03	5.5541E+03	**5.5642E+03**	5.5215E+03
Weish08	5	40	5.6050E+03	5.3898E+03	5.4367E+03	5.4939E+03	5.5520E+03	5.5939E+03	**5.6037E+03**	5.5624E+03
Weish09	5	40	5.2460E+03	4.8578E+03	4.9662E+03	5.0887E+03	5.1661E+03	5.2280E+03	**5.2445E+03**	5.1912E+03
Weish10	5	50	6.3390E+03	5.6458E+03	5.7433E+03	5.9010E+03	6.0528E+03	6.2422E+03	**6.3333E+03**	6.2951E+03
Weish11	5	50	5.6430E+03	4.7359E+03	4.8966E+03	5.0982E+03	5.3064E+03	5.4929E+03	**5.6310E+03**	5.5763E+03
Weish12	5	50	6.3390E+03	5.6023E+03	5.7108E+03	5.8669E+03	6.0099E+03	6.2249E+03	**6.3347E+03**	6.2824E+03
Weish13	5	50	6.1590E+03	5.3972E+03	5.5157E+03	5.6411E+03	5.8484E+03	6.0470E+03	**6.1545E+03**	6.0931E+03
Weish14	5	60	6.9540E+03	5.8835E+03	6.0450E+03	6.1776E+03	6.3809E+03	6.6598E+03	**6.9289E+03**	6.9092E+03
Weish15	5	60	7.4860E+03	6.0940E+03	6.2858E+03	6.5758E+03	6.8966E+03	7.2223E+03	7.4580E+03	7.4594E+03
Weish16	5	60	7.2890E+03	6.3411E+03	6.4966E+03	6.6618E+03	6.8396E+03	7.0776E+03	**7.2769E+03**	7.2494E+03
Weish17	5	60	8.6330E+03	8.0783E+03	8.1598E+03	8.2361E+03	8.3520E+03	8.4967E+03	**8.6218E+03**	8.6152E+03
Weish18	5	70	9.5800E+03	8.6638E+03	8.7475E+03	8.8790E+03	9.0582E+03	9.2736E+03	9.5151E+03	**9.5340E+03**
Weish19	5	70	7.6980E+03	6.1447E+03	6.2935E+03	6.5528E+03	6.8217E+03	7.1677E+03	7.5972E+03	**7.6228E+03**
Weish20	5	70	9.4500E+03	8.0720E+03	8.2605E+03	8.4773E+03	8.7069E+03	9.0362E+03	9.3837E+03	**9.4075E+03**
Weish21	5	70	9.0740E+03	7.5268E+03	7.7440E+03	8.0049E+03	8.2594E+03	8.6040E+03	8.9957E+03	**9.0291E+03**
Weish22	5	80	8.9470E+03	7.0712E+03	7.2595E+03	7.4979E+03	7.7506E+03	8.1719E+03	8.7438E+03	**8.8596E+03**
Weish23	5	80	8.3440E+03	6.2538E+03	6.5068E+03	6.7881E+03	7.1018E+03	7.5522E+03	8.1184E+03	**8.2691E+03**
Weish24	5	80	1.0220E+04	9.0657E+03	9.1887E+03	9.3318E+03	9.5167E+03	9.7613E+03	1.0087E+04	**1.0176E+04**
Weish25	5	80	9.9390E+03	8.3582E+03	8.5302E+03	8.7430E+03	9.0253E+03	9.3687E+03	9.7981E+03	**9.9088E+03**
Weish26	5	90	9.5840E+03	7.1117E+03	7.4250E+03	7.6822E+03	8.0370E+03	8.5056E+03	9.2359E+03	**9.5094E+03**
Weish27	5	90	9.8190E+03	7.4241E+03	7.6435E+03	7.9445E+03	8.2766E+03	8.7483E+03	9.4363E+03	**9.7328E+03**
Weish28	5	90	9.4920E+03	6.9374E+03	7.2043E+03	7.5403E+03	7.9370E+03	8.4264E+03	9.1158E+03	**9.4159E+03**
Weish29	5	90	9.4100E+03	6.8325E+03	7.0784E+03	7.4160E+03	7.8180E+03	8.2937E+03	9.0303E+03	**9.3073E+03**
Weish30	5	90	1.1191E+04	9.5553E+03	9.7001E+03	9.9120E+03	1.0183E+04	1.0548E+04	1.1002E+04	**1.1160E+04**

Under the t-test with significance level of 0.05 and Bonferroni correction, if the mean of one result is significantly better than all the others, then its corresponding mean value is highlighted in bold.

where f^* is the optimal (maximal) result, and f_{mean} is the mean value obtained by the algorithm.

Since $\Delta\%$ is a normalized measure (between 0 and 1) across different instances, we calculated the mean $\Delta\%$ value for each MKP set and all the MKP sets as a whole. The results are shown in Table IV. For each dataset, the best value is marked in bold. It can be seen that the value of 0.9 obtained the smallest mean Δ% over all the instances, and it was outperformed by $w = 1.0$ only on the Weish set (Weish18 to Weish30 shown by Table III). Therefore, $w^* = 0.9$ is selected as the best constant inertia weight for "Con" in the subsequent empirical studies.

TABLE IV
MEAN Δ% VALUES OF "CON" WITH DIFFERENT INERTIA WEIGHT OVER 100 INDEPENDENT RUNS ON THE MKP DATASETS SEPARATELY AND ALTOGETHER. FOR EACH DATASET, THE BEST VALUE IS MARKED IN BOLD.

Dataset	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Sento	11.6%	9.5%	7.4%	4.9%	2.6%	**0.4%**	0.9%
Weing	11.2%	8.7%	6.7%	4.9%	3.3%	**1.2%**	2.5%
Weish	12.9%	11.2%	9.1%	6.8%	4.0%	1.0%	**0.8%**
Overall	12.5%	10.6%	8.5%	6.3%	3.8%	**1.0%**	1.1%

C. Experimental Results

In this section, we compare the "Down", "Con" and "Up" strategies. For each test instance and each compared algorithm, 100 independent runs were conducted, and the mean and standard deviation of the 100 corresponding results were calculated. For each instance, the best algorithm was compared with the other two algorithms by the t-test with significance level of 0.05 and Bonferroni correction. If the best algorithm is significantly better than the other two, then the corresponding entry is marked in bold. In addition, if an algorithm consistently achieved the optimal value (best known value for the SKP instances, where the optimal value is unknown), the corresponding entry is marked with *.

1) Single 0/1 Knapsack Problem: Table V shows the mean and standard deviation of the results of the compared algorithms over 100 independent runs on the SKP instances, along with the number of items n of each instance, which ranges from 8 to 24. Since the instances are randomly generated, the optimal solutions for them are unknown. The column "BK" gives the best known profit for each instance, which is the maximal profit obtained by all the tested algorithms.

From the table, one can see that "Up" performed significantly better than both "Down" and "Con" on 3 out of the total 25 instances. On another 9 instances (Ks_8a to Ks_8e, Ks_12a and Ks_12c to Ks_12e), "Up" consistently reached the best known solution along with at least one of the other algorithms. "Up" was not significantly outperformed on any SKP instance. Thus, the efficacy of the proposed increasing inertia weight scheme has been demonstrated on SKP instances.

2) Multi-dimensional 0/1 Knapsack Problem: The mean and standard deviation of the results of the compared algorithms over 100 independent runs on the MKP instances are given in Tables VI and VII. The column "Success Rate" indicates the percentage of the runs that successfully achieved the global optimum over the total 100 runs.

Table VI shows the results on the Sento and Weing instances. From the table, one can see that "Up" obtained significantly better results than "Down" and "Con" on 3 out of the total 10 instances, and was statistically comparable with the other algorithms on the remaining 7 instances. In terms of success rate, "Up" obtaining the highest success rate on 6 instances, including the last two instances for which "Down" and "Con" failed to find the global optimum.

Table VII shows the results on the Weish instances, in which similar patterns can be observed. In terms of both mean profit and success rate, "Up" performed better than both "Down" and "Con". The statistical test shows that "Up" performed significantly better than "Down" and "Con" on 16 out of the total 30 instances, and was only outperformed by "Con" on 1 instances. "Up" also achieved much higher success rate than the other algorithms on most of the instances. Table VIII shows that over all the MKP instances, the mean Δ% value of "Up" is 0.2%, which is much smaller than that of "Down" (0.5%) and "Con" (1.0%).

Note that Table III shows that the $w = 1.0$ is better than $w = 0.9$ on Weish18 to Weish30. To make a fairer comparison, we also conducted t-test with significance level of 0.05 between "Up" and "Con" with $w = 1.0$, and the results showed that "Up" still performed significantly better than "Con" with $w = 1.0$ on Weish18 to Weish30. This verifies the efficacy of "Up" on the MKP instances, especially on the large instances.

TABLE VIII
MEAN Δ% VALUES OF "DOWN", "CON" AND "UP" OVER 100 INDEPENDENT RUNS ON THE MKP DATASETS SEPARATELY AND ALTOGETHER. FOR EACH DATASET, THE BEST VALUE IS MARKED IN BOLD.

Dataset	Down	Con	Up
Sento	0.6%	0.4%	**0.3%**
Weing	0.7%	1.2%	**0.2%**
Weish	0.5%	1.0%	**0.2%**
Overall	0.5%	1.0%	**0.2%**

Fig. 5 shows the boxplots of the distribution of the Δ% values in the 100 independent runs of "Down", "Con" and "Up", on some representative MKP instances in terms of problem size. The plots clearly show that the distribution of "Up" are below that of both "Down" and "Con" for these instances. For Weish03 and Weish13, the median of "Up" reached zero, indicating that "Up" is more likely to achieve the optimal solutions than the other algorithms on these instances.

Note that when using the increasing inertia weight scheme in the experiments, w is increased from the lower bound $\underline{w}$ to the upper bound $\overline{w}$ during the first $\rho \cdot \overline{w}$ iterations. Therefore, the increasing rate of w depends on the total number of iterations $\overline{\pi}$. Specifically, when given less iterations, the increasing speed of w is higher, and the BPSO moves from exploration to exploitation more rapidly. Likewise, when adopting the decreasing inertia weight scheme, w is decreased with a higher speed under a smaller number of iterations. To investigate the effect of $\overline{\pi}$ on the behavior of the BPSO with the increasing inertia weight scheme, four different BPSO variants with maximal number of iterations of 500, 1000, 2000 and 3000 are tested on the Weish MKP instances and their convergence curves are compared.

Fig. 6 shows the convergence curves of the compared algorithms with the maximal number of iterations of 500, 1000,

TABLE V
THE RESULTS OF "DOWN", "CON" AND "UP" OVER INDEPENDENT 100 RUNS ON THE SKP INSTANCES.

Name	n	BK	Down	Con	Up
Ks_8a	8	3.9244E+06	3.9244E+06(0.0000E+00)*	3.9228E+06(4.1371E+03)	3.9244E+06(0.0000E+00)*
Ks_8b	8	3.8137E+06	3.8137E+06(0.0000E+00)*	3.8137E+06(0.0000E+00)*	3.8137E+06(0.0000E+00)*
Ks_8c	8	3.3475E+06	3.3475E+06(0.0000E+00)*	3.3458E+06(1.6384E+04)	3.3475E+06(0.0000E+00)*
Ks_8d	8	4.1877E+06	4.1877E+06(0.0000E+00)*	4.1877E+06(0.0000E+00)*	4.1877E+06(0.0000E+00)*
Ks_8e	8	4.9556E+06	4.9556E+06(0.0000E+00)*	4.9497E+06(2.3469E+04)	4.9556E+06(0.0000E+00)*
Ks_12a	12	5.6889E+06	5.6889E+06(0.0000E+00)*	5.6884E+06(3.1029E+03)	5.6889E+06(0.0000E+00)*
Ks_12b	12	6.4986E+06	6.4948E+06(9.1792E+03)	6.4902E+06(1.2088E+04)	6.4971E+06(6.1050E+03)
Ks_12c	12	5.1706E+06	5.1706E+06(0.0000E+00)*	5.1694E+06(6.4341E+03)	5.1706E+06(0.0000E+00)*
Ks_12d	12	6.9924E+06	6.9924E+06(0.0000E+00)*	6.9921E+06(1.0333E+03)	6.9924E+06(0.0000E+00)*
Ks_12e	12	5.3375E+06	5.3375E+06(0.0000E+00)*	5.3347E+06(1.7023E+04)	5.3375E+06(0.0000E+00)*
Ks_16a	16	7.8510E+06	7.8455E+06(9.8417E+03)	7.8382E+06(1.6408E+04)	**7.8488E+06(6.6265E+03)**
Ks_16b	16	9.3530E+06	9.3523E+06(3.1377E+03)	9.3509E+06(9.1346E+03)	9.3528E+06(9.0215E+02)
Ks_16c	16	9.1511E+06	9.1438E+06(1.5740E+04)	9.1350E+06(2.4331E+04)	**9.1492E+06(6.7132E+03)**
Ks_16d	16	9.3489E+06	9.3447E+06(7.4229E+03)	9.3370E+06(1.7258E+04)	9.3450E+06(7.7160E+03)
Ks_16e	16	7.7691E+06	7.7664E+06(6.3825E+03)	7.7630E+06(8.2523E+03)	7.7673E+06(5.2458E+03)
Ks_20a	20	1.0727E+07	1.0718E+07(1.3744E+04)	1.0720E+07(1.4229E+04)	1.0719E+07(1.3242E+04)
Ks_20b	20	9.8183E+06	9.7950E+06(2.4608E+04)	9.7978E+06(2.4051E+04)	9.8014E+06(2.3655E+04)
Ks_20c	20	1.0714E+07	1.0707E+07(1.5817E+04)	1.0705E+07(2.3827E+04)	1.0710E+07(5.8940E+03)
Ks_20d	20	8.9292E+06	8.9149E+06(1.9316E+04)	8.9160E+06(2.1221E+04)	8.9214E+06(1.3997E+04)
Ks_20e	20	9.3580E+06	9.3522E+06(1.0796E+04)	9.3518E+06(1.1703E+04)	**9.3550E+06(5.7768E+03)**
Ks_24a	24	1.3549E+07	1.3512E+07(2.8550E+04)	1.3524E+07(2.2450E+04)	1.3521E+07(2.7622E+04)
Ks_24b	24	1.2234E+07	1.2199E+07(2.4597E+04)	1.2216E+07(1.9263E+04)	1.2214E+07(2.0903E+04)
Ks_24c	24	1.2449E+07	1.2423E+07(2.5462E+04)	1.2434E+07(1.9286E+04)	1.2434E+07(1.8010E+04)
Ks_24d	24	1.1815E+07	1.1793E+07(2.0646E+04)	1.1798E+07(1.7061E+04)	1.1797E+07(1.6720E+04)
Ks_24e	24	1.3940E+07	1.3921E+07(2.1878E+04)	1.3928E+07(1.5814E+04)	1.3923E+07(1.7231E+04)

Under the t-test with significance level of 0.05 and Bonferroni correction, if the best algorithm performed significantly better than all the others, then the corresponding entry is marked in bold. If an algorithm consistently achieved the maximal value, the corresponding entry is marked with *.

TABLE VI
THE RESULTS OF "DOWN", "CON" AND "UP" OVER 100 INDEPENDENT RUNS ON THE SENTO AND WEING MKP INSTANCES.

Name	m	n	Opt	Success Rate			Mean(Std)		
				Down	Con	Up	Down	Con	Up
Sento1	30	60	7.7720E+03	12	11	24	7.7095E+03(5.1993E+01)	7.7335E+03(2.5956E+01)	7.7422E+03(3.0506E+01)
Sento2	30	60	8.7220E+03	1	4	6	8.6889E+03(2.3588E+01)	8.6948E+03(1.5545E+01)	8.6992E+03(1.7186E+01)
Weing1	2	28	1.4128E+05	92	98	100	1.4126E+05(1.0175E+02)	1.4127E+05(8.8480E+01)	1.4128E+05(0.0000E+00)*
Weing2	2	28	1.3088E+05	94	100	90	1.3087E+05(3.8189E+01)	1.3088E+05(0.0000E+00)*	1.3087E+05(4.8242E+01)
Weing3	2	28	9.5677E+04	16	31	29	9.5162E+04(5.8707E+02)	9.5222E+04(4.6670E+02)	9.5340E+04(4.5517E+02)
Weing4	2	28	1.1934E+05	84	97	95	1.1904E+05(9.2077E+02)	1.1923E+05(6.0109E+02)	1.1932E+05(5.4542E+01)
Weing5	2	28	9.8796E+04	58	94	98	9.8102E+04(1.3940E+03)	9.8577E+04(8.7401E+02)	**9.8793E+04(2.3216E+01)**
Weing6	2	28	1.3062E+05	60	70	61	1.3044E+05(2.7742E+02)	1.3051E+05(1.7962E+02)	1.3047E+05(1.9118E+02)
Weing7	2	105	1.0954E+06	0	0	2	1.0916E+06(2.1860E+03)	1.0832E+06(3.0085E+03)	**1.0945E+06(1.2014E+03)**
Weing8	2	105	6.2432E+05	0	0	2	6.0384E+05(1.0683E+04)	5.7694E+05(1.1157E+04)	**6.1885E+05(2.2128E+03)**

Under the t-test with significance level of 0.05 and Bonferroni correction, if the best algorithm performed significantly better than all the others, the corresponding entry is marked in bold. If an algorithm consistently achieved the optimal value, the corresponding entry is marked with *.

2000 and 3000, on Weish03, Weish08, Weish13, Weish18, Weish23 and Weish28. They are six representative instances in the Weish dataset, which can cover the range of the problem sizes of all the Weish instances. In each sub-figure, the x-axis stands for the number of iterations, and the y-axis is the mean profit value of the best-so-far solutions over the 100 independent runs. For each sub-figure, an inner figure showing the zoom-in of the convergence curves is given to make a clearer observation of the difference between them.

Clearly, the convergence curve of "Con" is not affected by the maximal number of iterations, since w remains the same throughout the search process. For both "Up" and "Down", it is obvious that a larger number of iterations allowed a more exhaustive search within the solution space, and thus led to a better final solution. When comparing between "Up" and "Down", one can see that "Up" converged more slowly than "Down", but finally reached a better solution than "Down". This is consistent with the motivation of adopting the increasing inertia weight scheme. The convergence rate should be slow during the early exploration stage, and increase over time as the search is moving towards exploitation. The outperformance of "Up" over "Down" for all the tested numbers of iterations shows that the maximal number of iterations does not affect much the performance of the increasing inertia weight scheme. Additionally, "Up" performed better than "Con" on all the instances. However, the gap reduces as the problem size increases. This is because that a stronger exploitation capability of the BPSO is more desirable in a larger solution space.

In order to illustrate the effect of the inertia weight on the

TABLE VII

THE RESULTS OF "DOWN", "CON" AND "UP" OVER 100 INDEPENDENT RUNS ON THE WEISH MKP INSTANCES.

Name	m	n	Opt	Success Rate			Mean(Std)		
				Down	Con	Up	Down	Con	Up
Weish01	5	30	4.5540E+03	84	96	96	4.5455E+03(2.3441E+01)	4.5510E+03(1.5019E+01)	4.5510E+03(1.4873E+01)
Weish02	5	30	4.5360E+03	71	78	74	4.5302E+03(1.4955E+01)	4.5349E+03(2.0817E+00)	4.5336E+03(6.4193E+00)
Weish03	5	30	4.1150E+03	60	91	92	4.0937E+03(2.9322E+01)	4.1093E+03(1.8120E+01)	4.1108E+03(1.5166E+01)
Weish04	5	30	4.5610E+03	100	100	100	4.5610E+03(0.0000E+00)*	4.5610E+03(0.0000E+00)*	4.5610E+03(0.0000E+00)*
Weish05	5	30	4.5140E+03	100	100	100	4.5140E+03(0.0000E+00)*	4.5140E+03(0.0000E+00)*	4.5140E+03(0.0000E+00)*
Weish06	5	40	5.5570E+03	33	55	61	5.5389E+03(1.9317E+01)	5.5498E+03(8.3113E+00)	5.5502E+03(9.0321E+00)
Weish07	5	40	5.5670E+03	61	85	75	5.5563E+03(1.9244E+01)	5.5642E+03(6.6619E+00)	5.5617E+03(9.6944E+00)
Weish08	5	40	5.6050E+03	34	60	38	5.5947E+03(2.0009E+01)	**5.6037E+03(2.8476E+00)**	5.6017E+03(6.0944E+00)
Weish09	5	40	5.2460E+03	83	96	93	5.2387E+03(1.6689E+01)	5.2445E+03(7.3615E+00)	5.2434E+03(9.7055E+000
Weish10	5	50	6.3390E+03	45	69	65	6.3242E+03(2.2378E+01)	6.3333E+03(1.4700E+01)	6.3343E+03(1.4201E+01)
Weish11	5	50	5.6430E+03	26	54	38	5.5966E+03(5.2928E+01)	5.6310E+03(2.3622E+01)	5.6222E+03(3.3299E+01)
Weish12	5	50	6.3390E+03	51	76	62	6.3189E+03(3.7169E+01)	6.3347E+03(1.4587E+01)	6.3346E+03(1.4979E+01)
Weish13	5	50	6.1590E+03	60	90	94	6.1304E+03(4.3249E+01)	6.1545E+03(1.5011E+01)	6.1547E+03(1.9201E+01)
Weish14	5	60	6.9540E+03	61	44	74	6.9304E+03(4.1637E+01)	6.9289E+03(2.7549E+01)	**6.9429E+03(2.0835E+01)**
Weish15	5	60	7.4860E+03	50	27	73	7.4623E+03(2.7134E+01)	7.4580E+03(2.1533E+01)	**7.4751E+03(1.9602E+01)**
Weish16	5	60	7.2890E+03	14	23	26	7.2621E+03(3.4636E+01)	7.2769E+03(1.4350E+01)	7.2793E+03(1.5876E+01)
Weish17	5	60	8.6330E+03	30	23	41	8.6222E+03(1.0012E+01)	8.6218E+03(8.4226E+00)	**8.6254E+03(8.0533E+00)**
Weish18	5	70	9.5800E+03	7	0	24	9.5456E+03(2.6380E+01)	9.5151E+03(2.8563E+01)	**9.5605E+03(1.6840E+01)**
Weish19	5	70	7.6980E+03	19	4	45	7.6461E+03(4.8230E+01)	7.5972E+03(5.4987E+01)	**7.6786E+03(2.7084E+01)**
Weish20	5	70	9.4500E+03	26	2	38	9.4220E+03(3.7139E+01)	9.3837E+03(3.2992E+01)	**9.4379E+03(1.3907E+01)**
Weish21	5	70	9.0740E+03	16	2	48	9.0314E+03(3.2294E+01)	8.9957E+03(3.7999E+01)	**9.0546E+03(2.3079E+01)**
Weish22	5	80	8.9470E+03	10	0	25	8.8702E+03(5.0240E+01)	8.7438E+03(6.9980E+01)	**8.9146E+03(2.9141E+01)**
Weish23	5	80	8.3440E+03	3	0	11	8.2603E+03(5.5769E+01)	8.1184E+03(7.8012E+01)	**8.3136E+03(3.5389E+01)**
Weish24	5	80	1.0220E+04	3	0	27	1.0164E+04(3.0367E+01)	1.0087E+04(4.1167E+01)	**1.0198E+04(2.3412E+01)**
Weish25	5	80	9.9390E+03	1	0	19	9.8998E+03(2.1386E+01)	9.7981E+03(4.4962E+01)	**9.9191E+03(1.3407E+01)**
Weish26	5	90	9.5840E+03	4	0	15	9.4756E+03(6.5445E+01)	9.2359E+03(8.7741E+01)	**9.5471E+03(2.5547E+01)**
Weish27	5	90	9.8190E+03	8	0	55	9.6903E+03(8.4437E+01)	9.4363E+03(8.3036E+01)	**9.7852E+03(5.5456E+01)**
Weish28	5	90	9.4920E+03	4	0	45	9.3968E+03(6.9946E+01)	9.1158E+03(9.0954E+01)	**9.4647E+03(3.5504E+01)**
Weish29	5	90	9.4100E+03	2	0	21	9.2880E+03(6.4943E+01)	9.0303E+03(8.3109E+01)	**9.3640E+03(3.7572E+01)**
Weish30	5	90	1.1191E+04	2	0	25	1.1137E+04(2.5216E+01)	1.1002E+04(4.6196E+01)	**1.1174E+04(1.5945E+01)**

Under the t-test with significance level of 0.05 and Bonferroni correction, if the best algorithm performed significantly better than all the others, the corresponding entry is marked in bold. If an algorithm consistently achieved the optimal value, the corresponding entry is marked with *.

diversity of BPSO during the search process, we calculated the curves of the mean mutual distance between the particles during the search process of "Up", "Down" and "Con". Fig. 7 shows such a diversity plot on Weish18. The other instances showed a similar pattern. It is clear that "Con" has a standard search process, in which the diversity drops rapidly and then converges to a certain level (around 17). In "Up", the mutual distance between the particles decreases much slower, which indicates that the search takes more effort in exploration than "Con". In "Down", the mutual distance between the particles first decreases at the very early stage of the search, since all the particles move towards some local optima. Then, the mutual distance starts to increase, implying that the search shifts from exploitation to exploration again. This phenomenon is consistent with our discussions that the exploration capability of BPSO increases with the decrease of the inertia weight.

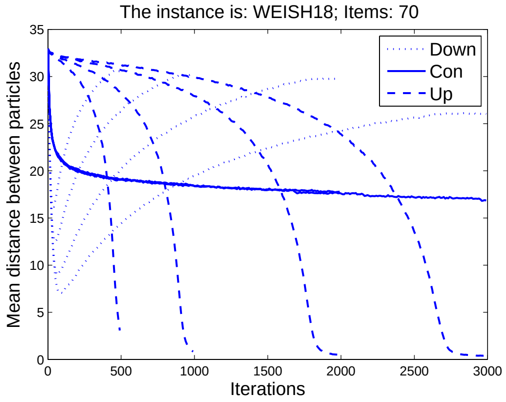
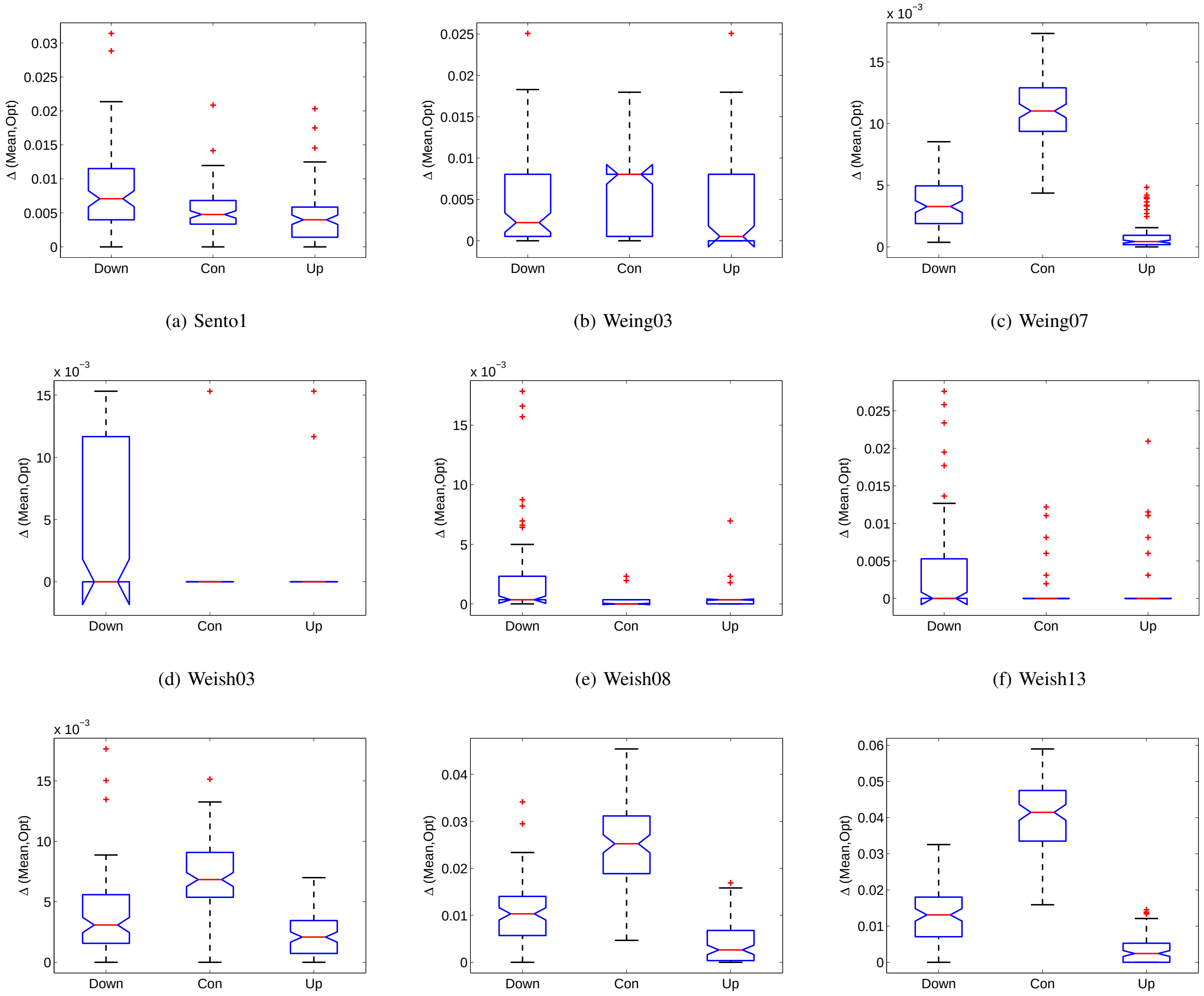


Fig. 7. The curves of the mean mutual distance between the particles during the search process of "Up", "Down" and "Con" with different maximal fitness evaluations on Weish18.

VI. CONCLUSION

In this paper, the effect of the inertia weight on the behavior of BPSO is investigated both theoretically and empirically. First, the behavior of BPSO is theoretically analyzed under different scenarios and inertia weight settings, and it was discovered that under the assumption of stagnation (the global and personal best positions are unchanged over time) and when the acceleration coefficients c_1 and c_2 are fixed, a larger inertia weight w tends to encourage exploitation, while a smaller one tends to lead to a strong exploration capability.

Based on the above observations and the intuition that the search should start with exploration, and then gradually move towards exploitation, a linearly increasing inertia weight scheme is proposed for BPSO. The experimental studies on the 0/1 knapsack problem demonstrate that the proposed increasing inertia weight scheme performed much better than the decreasing and constant counterparts of BPSO for solving



(a) Sento1

(b) Weing03

(c) Weing07

(d) Weish03

(e) Weish08

(f) Weish13

(g) Weish18

(h) Weish23

(i) Weish28

Fig. 5. The boxplots of the Δ% values in the 100 independent runs of “Up”, “Down” and “Con” on some representative MKP instances.

binary optimization problems.

In the future, we will extend the analysis from a single bit to a bit string, based on which more sophisticated inertia weight updating schemes will be designed. Currently, the w value is unique for all the bits of the bit string, regardless of their own situations, e.g. whether the bit values of the personal and global best positions are the same. Furthermore, an inertia weight vector $\mathbf{w}$ could be defined, so that each bit can have its own inertia weight setting based on the particular desirable tradeoff between exploration and exploitation for that bit.

References

[1] J. Kennedy, R. Eberhart *et al.*, “Particle swarm optimization,” in *Proceedings of IEEE international conference on neural networks*, vol. 4, no. 2. Perth, Australia, 1995, pp. 1942–1948.

[2] J. Kennedy, “Particle swarm optimization,” in *Encyclopedia of Machine Learning*. Springer, 2010, pp. 760–766.

[3] W.-N. Chen, J. Zhang, H. S.-H. Chung, W.-L. Zhong, W.-G. Wu, and Y.-H. Shi, “A novel set-based particle swarm optimization method for discrete optimization problems,” *Evolutionary Computation, IEEE Transactions on*, vol. 14, no. 2, pp. 278–300, 2010.

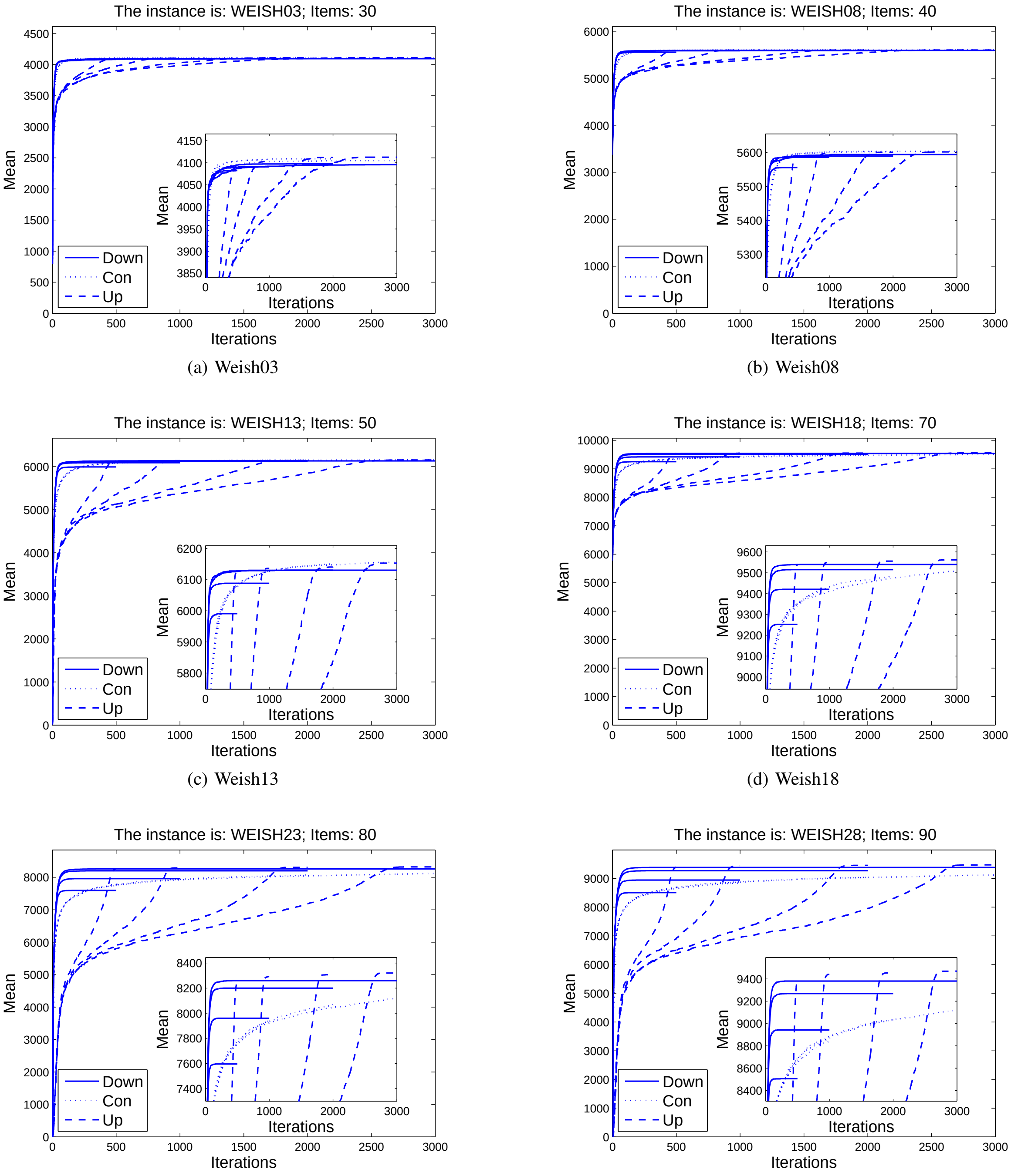
[4] B. Jarboui, N. Damak, P. Siarry, and A. Rebai, “A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems,” *Applied Mathematics and Computation*, vol. 195, no. 1, pp. 299–308, 2008.

[5] J. Kennedy and R. C. Eberhart, “A discrete binary version of the particle swarm algorithm,” in *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on*, vol. 5. IEEE, 1997, pp. 4104–4108.

[6] G. Pampara, N. Franken, and A. P. Engelbrecht, “Combining particle swarm optimisation with angle modulation to solve binary problems,” in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, vol. 1. IEEE, 2005, pp. 89–96.

[7] M. Gong, Q. Cai, X. Chen, and L. Ma, “Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition,” *Evolutionary Computation, IEEE Transactions on*, vol. 18, no. 1, pp. 82–97, 2014.

[8] Y.-J. Zheng, H.-F. Ling, J.-Y. Xue, and S.-Y. Chen, “Population classification in fire evacuation: a multiobjective particle swarm optimization approach,” *Evolutionary Computation, IEEE Transactions on*, vol. 18, no. 1, pp. 70–81, 2014.



(a) Weish03

(b) Weish08

(c) Weish13

(d) Weish18

(e) Weish23

(f) Weish28

Fig. 6. The convergence curves of "Up", "Down" and "Con" with the maximal number of iterations of 500, 1000, 2000 and 3000, on a representative subset of Weish MKP instances.

[9] W. Hu and G. G. Yen, "Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system," *Evolutionary Computation, IEEE Transactions on*, vol. 19, no. 1, pp. 1–18, 2015.

[10] C.-J. Liao, C.-T. Tseng, and P. Luarn, "A discrete version of particle swarm optimization for flowshop scheduling problems," *Computers & Operations Research*, vol. 34, no. 10, pp. 3099–3111, 2007.

[11] J. C. Bansal and K. Deep, "A modified binary particle swarm optimization for knapsack problems," *Applied Mathematics and Computation*, vol. 218, no. 22, pp. 11 042–11 061, 2012.

[12] A. Ahmadi, Y. Alinejad-Beromi, and M. Moradi, "Optimal pmu placement for power system observability using binary particle swarm optimization and considering measurement redundancy," *Expert Systems with Applications*, vol. 38, no. 6, pp. 7263–7269, 2011.

[13] S. M. Vieira, L. F. Mendonça, G. J. Farinha, and J. Sousa, "Modified binary pso for feature selection using svm applied to mortality prediction of septic patients," *Applied Soft Computing*, vol. 13, no. 8, pp. 3494–3504, 2013.

[14] Y. Zhang, S. Wang, P. Phillips, and G. Ji, "Binary pso with mutation operator for feature selection using decision tree applied to spam detection," *Knowledge-Based Systems*, vol. 64, pp. 22–31, 2014.

[15] M. G. Omran, A. Salman, and A. P. Engelbrecht, "Dynamic clustering using particle swarm optimization with application in image segmentation," *Pattern Analysis and Applications*, vol. 8, no. 4, pp. 332–344, 2006.

[16] J. Yang, H. Zhang, Y. Ling, C. Pan, and W. Sun, “Task allocation for wireless sensor network using modified binary particle swarm optimization,” 2014.

[17] A. H. El-Maleh, A. T. Sheikh, and S. M. Sait, “Binary particle swarm optimization (bpso) based state assignment for area minimization of sequential circuits,” *Applied Soft Computing*, vol. 13, no. 12, pp. 4832–4840, 2013.

[18] B. Wei and J. Zhao, “Haplotype inference using a novel binary particle swarm optimization algorithm,” *Applied Soft Computing*, vol. 21, pp. 415–422, 2014.

[19] A. Chatterjee, G. Mahanti, and A. Mahanti, “Synthesis of thinned concentric ring array antenna in predefined phi-planes using binary firefly and binary particle swarm optimization algorithm,” *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, 2014.

[20] C. Blum and X. Li, *Swarm intelligence in optimization*. Springer, 2008.

[21] I. C. Trelea, “The particle swarm optimization algorithm: convergence analysis and parameter selection,” *Information processing letters*, vol. 85, no. 6, pp. 317–325, 2003.

[22] M. Clerc and J. Kennedy, “The particle swarm-explosion, stability, and convergence in a multidimensional complex space,” *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 1, pp. 58–73, 2002.

[23] F. van den Bergh and A. P. Engelbrecht, “A study of particle swarm optimization particle trajectories,” *Information sciences*, vol. 176, no. 8, pp. 937–971, 2006.

[24] V. Kadirkamanathan, K. Selvarajah, and P. J. Fleming, “Stability analysis of the particle dynamics in particle swarm optimizer,” *Evolutionary Computation, IEEE Transactions on*, vol. 10, no. 3, pp. 245–255, 2006.

[25] J. L. Fernandez-Martinez and E. Garcia-Gonzalo, “Stochastic stability analysis of the linear continuous and discrete pso models,” *Evolutionary Computation, IEEE Transactions on*, vol. 15, no. 3, pp. 405–423, 2011.

[26] C. W. Cleghorn and A. P. Engelbrecht, “A generalized theoretical deterministic particle swarm model,” *Swarm Intelligence*, vol. 8, no. 1, pp. 35–59, 2014.

[27] Q. Liu, “Order-2 stability analysis of particle swarm optimization,” *Evolutionary computation*, vol. 23, no. 2, pp. 187–216, 2014.

[28] M. r. Bonyadi and Z. Michalewicz, “Analysis of stability, local convergence, and transformation sensitivity of a variant of particle swarm optimization algorithm,” *IEEE Transactions on Evolutionary Computation*, 2015. [Online]. Available: http://dx.doi.org/10.1109/TEVC.2015.2460753

[29] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*. Wiley, 2005.

[30] Y. Shi and R. Eberhart, “A modified particle swarm optimizer,” in *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*. IEEE, 1998, pp. 69–73.

[31] K. E. Parsopoulos and M. N. Vrahatis, “Recent approaches to global optimization problems through particle swarm optimization,” *Natural computing*, vol. 1, no. 2-3, pp. 235–306, 2002.

[32] Y. Shi and R. C. Eberhart, “Empirical study of particle swarm optimization,” in *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, vol. 3. IEEE, 1999.

[33] R. Brits, A. P. Engelbrecht, and F. van den Bergh, “Locating multiple optima using particle swarm optimization,” *Applied Mathematics and Computation*, vol. 189, no. 2, pp. 1859–1883, 2007.

[34] Y. Shi and R. C. Eberhart, “Parameter selection in particle swarm optimization,” in *Evolutionary Programming VII*. Springer, 1998, pp. 591–600.

[35] J. F. Schutte and A. A. Groenwold, “A study of global optimization using particle swarms,” *Journal of Global Optimization*, vol. 31, no. 1, pp. 93–108, 2005.

[36] R. Poli, J. Kennedy, and T. Blackwell, “Particle swarm optimization,” *Swarm intelligence*, vol. 1, no. 1, pp. 33–57, 2007.

[37] Y. Zheng, L. Ma, L. Zhang, and J. Qian, “On the convergence analysis and parameter selection in particle swarm optimization,” in *Machine Learning and Cybernetics, 2003 International Conference on*, vol. 3. IEEE, 2003, pp. 1802–1807.

[38] Y. Shi and R. C. Eberhart, “Empirical study of particle swarm optimizer with an increasing inertia weight,” in *Evolutionary Computation, 2003. CEC’03. The 2003 Congress on*, vol. 1. IEEE, 2003, pp. 221–226.

[39] X. Hu, Y. Shi, and R. C. Eberhart, “Recent advances in particle swarm.” in *IEEE congress on evolutionary computation*, vol. 1, 2004, pp. 90–97.

[40] K. E. Parsopoulos, E. I. Papageorgiou, P. Groumpos, and M. N. Vrahatis, “A first study of fuzzy cognitive maps learning using particle swarm optimization,” in *Evolutionary Computation, 2003. CEC’03. The 2003 Congress on*, vol. 2. IEEE, 2003, pp. 1440–1447.

[41] A. Nickabadi, M. M. Ebadzadeh, and R. Safabakhsh, “A novel particle swarm optimization algorithm with adaptive inertia weight,” *Applied Soft Computing*, vol. 11, no. 4, pp. 3658–3670, 2011.

[42] M. A. Arasomwan and A. O. Adewumi, “On the performance of linear decreasing inertia weight particle swarm optimization for global optimization,” *The Scientific World Journal*, vol. 2013, 2013.

[43] S. Mirjalili and A. Lewis, “S-shaped versus v-shaped transfer functions for binary particle swarm optimization,” *Swarm and Evolutionary Computation*, vol. 9, pp. 1–14, 2013.

[44] X. Jin, J. Zhao, Y. Sun, K. Li, and B. Zhang, “Distribution network reconfiguration for load balancing using binary particle swarm optimization,” in *Power System Technology, 2004. PowerCon 2004. 2004 International Conference on*, vol. 1. IEEE, 2004, pp. 507–510.

[45] A. Unler and A. Murat, “A discrete particle swarm optimization method for feature selection in binary classification problems,” *European Journal of Operational Research*, vol. 206, no. 3, pp. 528–539, 2010.

[46] S. Pookpunt and W. Ongsakul, “Optimal placement of wind turbines within wind farm using binary particle swarm optimization with time-varying acceleration coefficients,” *Renewable Energy*, vol. 55, pp. 266–276, 2013.

[47] N. Jin and Y. Rahmat-Samii, “Advances in particle swarm optimization for antenna designs: Real-number, binary, single-objective and multiobjective implementations,” *Antennas and Propagation, IEEE Transactions on*, vol. 55, no. 3, pp. 556–567, 2007.

[48] M. A. Khanesar, M. Teshnehlab, and M. A. Shoorehdeli, “A novel binary particle swarm optimization,” in *Control & Automation, 2007. MED’07. Mediterranean Conference on*. IEEE, 2007, pp. 1–6.

[49] L.-Y. Chuang, C.-H. Yang, and J.-C. Li, “Chaotic maps based on binary particle swarm optimization for feature selection,” *Applied Soft Computing*, vol. 11, no. 1, pp. 239–248, 2011.

[50] F. Shahzad, A. R. Baig, S. Masood, M. Kamran, and N. Naveed, “Opposition-based particle swarm optimization with velocity clamping (ovcpso),” in *Advances in Computational Intelligence*. Springer, 2009, pp. 339–348.

[51] P. C. Chu and J. E. Beasley, “A genetic algorithm for the multidimensional knapsack problem,” *Journal of heuristics*, vol. 4, no. 1, pp. 63–86, 1998.

[52] C.-Y. Lee, Z.-J. Lee, and S.-F. Su, “A new approach for solving 0/1 knapsack problem,” in *Systems, Man and Cybernetics, 2006. SMC’06. IEEE International Conference on*, vol. 4. IEEE, 2006, pp. 3138–3143.

[53] J. Langeveld and A. P. Engelbrecht, “Set-based particle swarm optimization applied to the multidimensional knapsack problem,” *Swarm Intelligence*, vol. 6, no. 4, pp. 297–342, 2012.

[54] M. A. K. Azad, A. M. A. Rocha, and E. M. Fernandes, “Improved binary artificial fish swarm algorithm for the 0–1 multidimensional knapsack problems,” *Swarm and Evolutionary Computation*, vol. 14, pp. 66–75, 2014.

[55] D. Zou, L. Gao, S. Li, and J. Wu, “Solving 0–1 knapsack problem by a novel global harmony search algorithm,” *Applied Soft Computing*, vol. 11, no. 2, pp. 1556–1564, 2011.