

Dynamic search control-based particle swarm optimization for project scheduling problems

Ruey-Maw Chen¹ and Yin-Mou Shen²

Abstract

Many machinery manufacturings are categorized as multi-mode resource-constrained project scheduling problems which have attracted significant interest in recent years. It has been shown that such problems are non-deterministic polynomial-time-hard. Particle swarm optimization is one of the most commonly used metaheuristic. Multi-mode resource-constrained project scheduling problems comprise two sub-problems, namely, an activity operating priority and an activity operating mode sub-problems; hence, two particle swarm optimizations are used to solve these two sub-problems. In solving the activity priority sub-problem, a designed global guidance ratio is involved to control the particle's search behavior. Restated, guiding a diversification search at the beginning stage and conducting an intensification search at latter stage are controlled by adjusting the global guidance ratio. The particle swarm optimization combined with the global guidance ratio mechanism is named global guidance ratio–particle swarm optimization herein. Meanwhile, a non-fixed global guidance ratio adjustment is also suggested to further enhance the search performance. Moreover, different communication topologies for balancing the convergence of using global and local topologies are also suggested in global guidance ratio–particle swarm optimization to further improve the search efficiency. The performance of the proposed global guidance ratio–particle swarm optimization scheme is evaluated by solving all the multi-mode resource-constrained project scheduling problem instances in Project Scheduling Problem Library. It is shown that the scheduling solutions are in good agreement with those presented in the literatures. Hence, the effectiveness of the proposed global guidance ratio–particle swarm optimization scheme is confirmed.

Keywords

Optimization, particle swarm optimization, scheduling, multi-mode resource-constrained project scheduling problem, global guidance ratio

Date received: 7 September 2015; accepted: 8 March 2016

Academic Editor: Stephen D Prior

Introduction

Scheduling problems are common throughout the business, production, and management fields. Of the various scheduling problems faced by enterprises, multi-mode resource-constrained project scheduling problems (MRCPSPs) are one of the most common, for example, many machinery manufacturings are categorized as MRCPSPs. The aim in solving an MRCPSP is to determine the optimal activity sequence and operation mode which minimizes the makespan while simultaneously satisfying all the constraints imposed on the activity

precedence and resource allocation decisions. In other words, MRCPSPs comprise two sub-problems, namely,

¹Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung, Taiwan, ROC

²Department of Information Management, Kun Shan University, Tainan, Taiwan, ROC

Corresponding author:

Ruey-Maw Chen, Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, No.57, Sec. 2, Zhongshan Rd., Taiping Dist., Taichung 41170, Taiwan, ROC.
Email: raymond@mail.ncut.edu.tw



Creative Commons CC-BY: This article is distributed under the terms of the Creative Commons Attribution 3.0 License

(<http://www.creativecommons.org/licenses/by/3.0/>) which permits any use, reproduction and distribution of the work without

further permission provided the original work is attributed as specified on the SAGE and Open Access pages (<https://us.sagepub.com/en-us/nam/open-access-at-sage>).

an activity precedence problem and an activity operating mode problem. The activity precedence problem involves determining the sequence in which the activities should be performed so as to minimize the make-span, while the activity operating mode problem involves determining the mode in which each activity should be performed. MRCPSP differs from the classical resource-constrained project scheduling problem (RCPSPP) in that each activity can be executed in one of several different modes, where each mode varies in terms of both the resources required to execute the activity and the time required for activity completion. An MRCPSP is subject to two limitations, namely, precedence and resource. Furthermore, the resources may be either reusable (i.e. available in limited quantities during each time period) or non-reusable (i.e. limited for the entire project duration). Many studies have shown that MRCPSPs are non-deterministic polynomial-time (NP)-hard¹ and are thus not easily solved within a short period of time, particularly when the problem involves a large number of activities and/or resources. Therefore, while exact methods are available for obtaining optimal solutions for small-scale problems, metaheuristic algorithms are required as the scale and complexity of the problem increase. The literature contains many metaheuristic algorithms suitable for solving MRCPSPs, including ant colony optimization,² scatter search algorithms,³ genetic algorithms,⁴ particle swarm optimization (PSO),^{5,6} bee colony optimization,⁷ simulated annealing,⁸ and evolutionary algorithms.⁹

PSO, inspired by the social behavior of birds flocking or fish schooling, has been applied to a wide variety of problems in recent years, including traveling salesman problems (TSPs),¹⁰ vehicle routing problems (VRPs),^{11,12} RCPSPPs,¹³ and so on. PSO has very few parameters to adjust and tends to be both faster and cheaper than other optimization methods. As a result, it provides an attractive approach for solving MRCPSPs. As described above, an MRCPSP comprises an activity precedence problem and an activity operating mode problem. In this study, both problems are solved using PSO. However, the activity precedence problem is solved using the standard or conventional PSO method, while the activity operating mode problem is solved using the discrete PSO method.

When seeking optimal solutions in a large solution space, a search strategy which starts from a wide-scope search and then tends toward a smaller-scope search is required. Intrinsically, a PSO particle movement strategy driven by global experience facilitates a wide-scope search and enables rapid convergence to the local optimal search space. In contrast, a movement strategy driven by the individual experiences of the particles is more conducive to a smaller-scope search, but results in slower convergence to the global optimal solution. To ensure that the conventional and standard PSO

algorithms converge stably to the global (rather than local) optimal solution, this study uses six different linear or non-linear global guidance ratio (GGR) mechanisms, and four different communication topologies (global, lbest, random-link, and group¹⁴), to achieve a trade-off between the individual experience of the particles and the global experience of the swarm when updating the particle velocity during the PSO solution procedure. For convenience, the PSO methods combined with these GGR mechanisms are referred to as GGR-PSO schemes herein. The performance of the proposed GGR-PSO scheme is evaluated by solving all the MRCPSP cases in the Project Scheduling Problem Library (PSPLIB).¹⁵ It is shown that the scheduling solutions obtained for each case are in good agreement with the optimal (or best) outcomes presented in the literature.

The remainder of this article is organized as follows. Section "MRCPSP" describes the MRCPSP scheduling problem considered in this study. Section "PSO" introduces the PSO algorithms used to solve the two MRCPSP sub-problems. Section "GGR-PSO" presents the proposed GGR-PSO schemes. Section "Experimental results" presents and discusses the experimental results. Finally, section "Conclusion" provides some brief concluding remarks.

MRCPSP

An MRCPSP involves both an activity precedence decision and an activity operating mode decision. Notably, each activity has different operating modes, and each mode incurs different amounts of time and resources. The aim in solving an MRCPSP is thus to determine the activity sequence and resource allocation which allows the project to be completed within the shortest possible time subject to the imposed resource constraints. An MRCPSP can be more formally defined as follows:

1. An MRCPSP comprises n physical activities and two virtual activities marking the beginning and end of the project, respectively.
2. Some of the activities in an MRCPSP involve precedence. That is, some activities cannot be executed until other activities have first been completed.
3. Each activity in an MRCPSP can be executed in multiple modes, that is, $Mode_j = \{M_1, M_2, \dots, M_j\}$, where M_j denotes the number of possible operating modes for activity j .
4. The resources in an MRCPSP include both renewable resources and non-renewable resources. Renewable resources are available in limited quantities during each time period,

whereas non-renewable resources are limited over the duration of the project. Suppose that there are k numbers of types of resource available for a project. In other words, $R = \{R_1, R_2, \dots, R_k\}$ denotes the total set of reusable resources available and R_k denotes the aggregate amount of k -type resource available. The supply of reusable resources within each time unit is fixed. Therefore, during each time unit, the total amount of reusable resources needed to execute all the scheduled activities may not exceed the fixed aggregate amount of resources available. Similarly, assume that q numbers of non-reusable resources are available for the project. Thus, $N = \{N_1, N_2, \dots, N_q\}$ denotes the total set of non-reusable resources available and N_q denotes the aggregate amount of q -type resource available. Throughout the project, the amount of non-reusable resources is fixed. Hence, the total amount of non-reusable resources applied cannot exceed the total amount of non-reusable resources available throughout the project.

5. In an MRCPSP, the amount of reusable and non-reusable resources needed varies in accordance with the operating mode assigned to each activity. Suppose that the set of reusable resources required to complete all the activities is denoted as $\{r_{j,M_j}^1, \dots, r_{j,M_j}^k\}$, in which r_{j,M_j}^k is the amount of k -type resource required for the M_j operating mode of activity j . Similarly, suppose that the set of non-reusable resources required to complete all the activities is denoted as $\{n_{j,M_j}^1, \dots, n_{j,M_j}^q\}$, in which n_{j,M_j}^q is the amount of q -type resource needed for the M_j operating mode of activity j . Thus, satisfaction of the reusable resource constraint can be represented as $\sum_{j \in act} r_{j,M_j}^k \leq R_k$.

Similarly, satisfaction of the non-reusable resource constraint can be represented as $\sum_{j \in act} n_{j,M_j}^q \leq N_q$.

6. Any scheduling solution which satisfies both the activity precedence constraint and the resource (reusable and non-reusable) constraint is said to be feasible, whereas any solution which fails to satisfy one (or both) of the constraint(s) is said to be infeasible.

PSO

In the GGR-PSO scheme proposed in this study, the activity priority problem is solved using the conventional or standard PSO method, while the activity operating mode problem is solved using the discrete PSO

method. The details of the three PSO schemes are provided in the following sections.

Conventional PSO

PSO was first introduced by Kennedy and Eberhart¹⁶ in 1995. In the PSO solution procedure, the position of each particle represents a potential solution to the problem of interest, and the particles move progressively through the solution space searching for the ideal position (i.e. the optimal solution). As shown in equation (1), the particle velocity used to determine the particle position in each iteration is updated in accordance with both the local experience of the particle and the global experience of the swarm. Suppose that the swarm comprises N particles located in D -dimensional space. The position of particle i ($i = 1, \dots, N$) is thus composed of D vectors, $X_{ij} = \{X_{i1}, \dots, X_{iD}\}$, where X_{ij} denotes the vector of particle i at position j . Furthermore, the velocity of particle i is given by $V_i = \{V_{i1}, \dots, V_{iD}\}$, while the individual (local) experience of particle i is represented as $L_i = \{L_{i1}, \dots, L_{iD}\}$. Finally, the global experience of the swarm is denoted as $Gbest = \{G_1, \dots, G_D\}$

$$\begin{aligned} V_{ij}^{new} &= \omega \times V_{ij}^{old} + c_1 \times r_1 \times (L_{ij} - X_{ij}) \\ &\quad + c_2 \times r_2 \times (G_j - X_{ij}) \\ X_{ij}^{new} &= X_{ij} + V_{ij}^{new} \end{aligned} \quad (1)$$

In equation (1), w is an inertia weight parameter used to govern the extent to which the original velocity V^{old} of the particle determines the new velocity V^{new} . Conventionally, w is assigned a value of 0.8. Moreover, c_1 and c_2 are the learning factors used to control the effects of the local experience and global experience on the new velocity, respectively. Finally, r_1 and r_2 are the random numbers with values in the range of 0–1 and are used to govern the motion of the particles toward the local best experience position and global best experience position, respectively, during the search process.

Standard PSO

The standard PSO method was proposed by Bratton and Kennedy¹⁷ in 2007 and includes a new parameter known as the constriction factor (χ) to balance the effects of global exploration and local exploitation on the search process. Compared to the inertia weight parameter used in the conventional PSO scheme, the constriction factor improves the stability of the search process in the solution space. Velocity updating is performed in accordance with

$$\begin{aligned} V_{ij}^{new} &= \chi \\ &\quad \times \left(V_{ij}^{old} + c_1 \times r_1 \times (L_{ij} - X_{ij}) + c_2 \times r_2 \times (G_j - X_{ij}) \right) \end{aligned} \quad (2)$$

where χ is typically assigned a value of 0.72984.

Discrete PSO

The discrete PSO method was proposed by Kennedy and Eberhart¹⁸ in 1997 and has been successfully applied to many optimization-type problems. In discrete PSO, the particle velocity is updated as

$$V_{ij}^{new} = V_{ij}^{old} + c_1 \times r_1 \times (L_{ij} - X_{ij}) + c_2 \times r_2 \times (G_j - X_{ij}) \Big|_{V_{max}}^{V_{max}} \quad (3)$$

The position vector (X_{ij}), individual best experience position vector (L_{ij}), and global best experience position (G_j) are expressed as either 0 or 1. Since r_1 and r_2 are real numbers with values between 0 and 1, the particle velocity is also a real number. Consequently, equation (1) cannot be applied to update the particle position. Eberhart and Kennedy reasoned that particles with a higher velocity V_{ij} have a greater chance of having a position vector X_{ij} equal to 1, while particles with a lower velocity V_{ij} have a greater chance of having a position vector X_{ij} equal to 0. Hence, having updated the particle velocity using equation (4), the new particle position, X_{ij} , is determined as

$$X_{ij}^{new} = \begin{cases} 1 & \gamma < S(V_{ij}^{new}) \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where $S(V_{ij}^{new})$ is a sigmoid function with the form shown in equation (5) and γ is a random number between 0 and 1

$$S(V_{ij}^{new}) = \frac{1}{1 + \exp(-V_{ij}^{new})} \quad (5)$$

To prevent $S(V_{ij}^{new})$ from lying too close to 0 or 1, a parameter V_{max} is applied to limit the range of the updated particle velocity (see equation (3)). V_{max} is typically assigned a value of 6.0, that is, V has a value between -6 and 6 .

GGR-PSO

Two different communication topologies are commonly used to determine the global best experience position G_j in equations (1) and (2), namely, a global topology and a local topology. For convenience, the value of G_j obtained using the global topology approach is denoted as *gbest* in this study, while that obtained using the local topology approach is denoted as *lbest*. In general, *gbest* tends to cause the PSO search process to converge prematurely to a local optimal solution. In contrast, *lbest* tends to slow the rate at which the search process converges. Accordingly, Chen⁶ proposed a rand-link topology approach based on a ring topology and some

additional random links. More specifically, for particle i , the rand-link topology combines the *lbest* topology (determined by X_{i+1} , X_i , X_{i-1}) and several randomly selected particles. In addition, Chen and Wu¹⁴ presented a group topology based on the *lbest* topology and an additional grouping concept. Notably, the rand-link topology and group topology both combine the respective advantages of the *gbest* and *lbest* paradigms.

Intrinsically, the global best experience G_j and local best experience L_{ij} guide the movement of the particles globally and locally, respectively. Restated, each particle's movement is self-guided or uni-directed by the particle's own experience (L_{ij}) or the swarm experience (G_j). Therefore, G_j and L_{ij} represent the characteristics of intensification guidance (smaller-scope) and diversification guidance (wide-scope), respectively. Intuitively, an ideal solution search strategy is one which starts from a wide-scope search and then tends toward a smaller-scope search. Hence, this study proposes a velocity update mechanism, designated as the GGR mechanism, which constrains the PSO search behavior from a wide-scope search toward a smaller-scope search as the search procedure continues by dynamically adjusting the relative effects of the individual best experience L_{ij} and global experience G_j , respectively, when updating the particle velocity in every iteration. More specifically, in each iteration, the GGR value is correlated with the global best guidance probability and updated using a curve function with an output value in the range of 0–1. The updated value of the GGR is then compared with a random value (*rand*) in the range of 0–1. If *rand* is greater than GGR, the particle velocity V_{ij}^{new} is updated in accordance with the individual best experience position L_{ij} . Conversely, if *rand* is less than or equal to GGR, the particle velocity is updated in accordance with the global best experience position G_j . In other words, the velocity update processes in the conventional and standard PSO schemes are performed in accordance with equations (6) and (7), respectively. For both schemes, the particle position is then updated using equation (8)

$$\begin{cases} V_{ij}^{new} = \omega V_{ij}^{old} + c_1 \times r_1 \times (L_{ij} - X_{ij}), & rand > GGR \\ V_{ij}^{new} = \omega V_{ij}^{old} + c_2 \times r_2 \times (G_j - X_{ij}), & rand \leq GGR \end{cases} \quad (6)$$

$$\begin{cases} V_{ij}^{new} = \chi \left(V_{ij}^{old} + c_1 \times r_1 \times (L_{ij} - X_{ij}) \right), & rand > GGR \\ V_{ij}^{new} = \chi \left(V_{ij}^{old} + c_2 \times r_2 \times (G_j - X_{ij}) \right), & rand \leq GGR \end{cases} \quad (7)$$

$$X_{ij}^{new} = X_{ij} + V_{ij}^{new} \quad (8)$$

Under the proposed scheme, each particle has a GGR probability of moving toward the global best experience probability and a $(1 - GGR)$ probability of

moving toward the local best experience probability. Hence, in the initial stage of the search process, the particle motion is governed mainly by the best experience of the individual particles, and thus many diverse candidate solutions for the optimization problem are obtained. However, as the search procedure continues, the particle motion is constrained increasingly by the global best experience of the swarm, and hence the particles tend to converge toward the most promising region of the solution space. Notably, the GGR mechanism is applied only in the activity operating priority problem (i.e. only in the conventional or standard PSO method), not in the activity operating mode problem (i.e. the discrete PSO method).

In this study, the GGR value in each iteration is determined using six different curves, namely, a linear increasing curve, two Sugeno increasing curves, an S-increasing curve, a dual S-increasing curve, and a sigmoid increasing curve. Note that the non-linear curves are chosen as the reverse of six existing decreasing curve functions, namely, the Sugeno function,¹⁹ S-decreasing curve function,²⁰ dual S-decreasing curve, and sigmoid decreasing curve.²¹ The related curve functions are given as follows

$$GGR_{iter} = \frac{iter}{iter_{max}} \quad (9)$$

$$GGR_{iter} = 1 - \frac{1 - \frac{iter}{iter_{max}}}{1 + s \times \frac{iter}{iter_{max}}} \quad (10)$$

$$GGR_{iter} = 1 - \frac{1}{1 + \left(\frac{iter}{60}\right)^5} \quad (11)$$

$$GGR_{iter} = \begin{cases} 0.5 - \frac{0.5}{1 + \left(\frac{iter}{60}\right)^5} & \text{if } iter < iter_{middle} \\ 1 - \frac{0.5}{1 + \left(\frac{iter - iter_{middle}}{60}\right)^5}, & \text{else} \end{cases} \quad (12)$$

$$GGR_{iter} = \frac{1}{1 + e^{-0.082608 \times (iter - 0.5 \times iter_{max})}} \quad (13)$$

where $iter$ is the number of iterations and $iter_{max}$ is the maximum number of iterations. Figure 1 presents a flowchart of the proposed GGR-PSO solution procedure.

Experimental results

The MRCPSP cases in PSPLIB comprise 10–30 jobs (activities) (denoted by J10, J12, J14, J16, J18, J20, and J30, respectively), where each activity has three different usable execution modes. Furthermore, each case

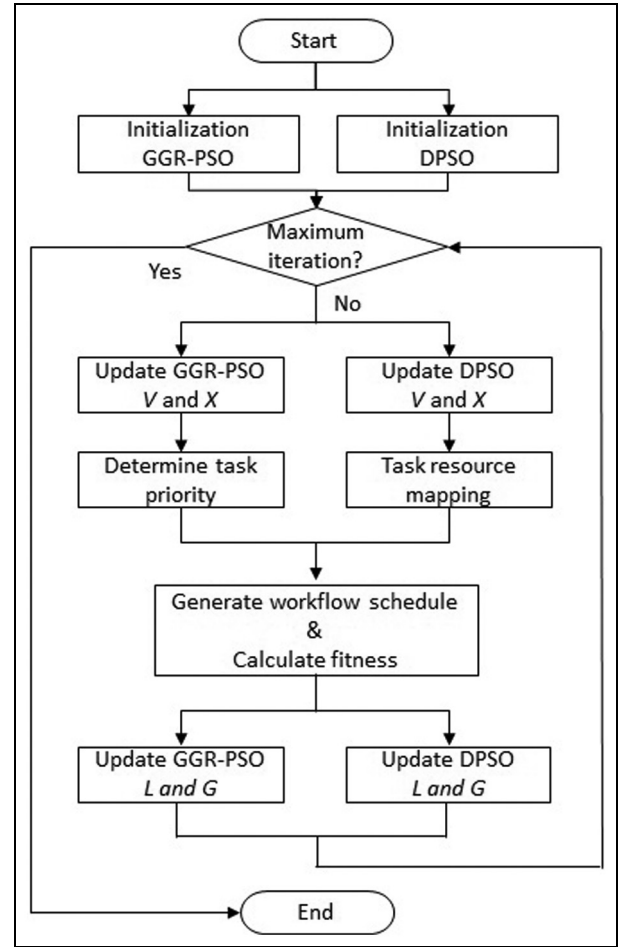


Figure 1. Flowchart of proposed GGR-PSO scheme.

has many different instances. Tables 1 and 2 show the precedence constraints and resource constraints, respectively, for one instance of the J12 case. In both tables, jobs 0 and 13 are pseudo jobs indicating the start and end of the project, respectively. Figure 2 illustrates the scheduling solution obtained by the proposed GGR-PSO scheme. (Note that the arrows represent the precedence relationships, and the labels N/M/T within each node represent the job number (N), execution mode (M), and finish time (T), respectively.)

The performance of the proposed GGR-PSO scheme was evaluated initially by solving all the instances of three MRCPSPs in the PSPLIB, namely, J10 (536 instances), J20 (554 instances), and J30 (552 instances). The performance of the GGR-PSO scheme was then compared with that of other state-of-the-art algorithms in the literature for the remaining MRCPSPs in PSPLIB. In implementing the proposed scheme, the learning factors in the velocity update equations were specified as $c_1 = c_2 = 2.0$, and the upper bound on the particle velocity in the discrete PSO scheme was set as $V_{max} = 6.0$. In every experiment, the aim was to produce 5000 feasible solutions for each instance in order

Table 1. Illustrative J12 instance (precedence constraints).

Job no.	No. of modes	No. of successors	Successors		
0	1	3	1	2	3
1	3	3	7	8	9
2	3	2	9	11	
3	3	3	4	5	6
4	3	2	10	12	
5	3	3	7	10	11
6	3	3	7	8	9
7	3	1	12		
8	3	2	10	11	
9	3	1	12		
10	3	1	13		
11	3	1	13		
12	3	1	13		
13	1	0			

Table 2. Illustrative J12 instance (resource constraints).

Job no.	Mode	Duration	R1	R2	N1	N2
0	1	0	0	0	0	0
1	1	1	8	4	6	5
	2	7	6	3	5	3
	3	8	4	2	5	3
2	1	7	3	7	6	7
	2	7	4	5	6	5
	3	10	1	2	5	2
3	1	2	7	9	6	7
	2	9	5	9	5	6
	3	10	3	8	4	6
4	1	5	5	8	10	9
	2	6	3	7	7	7
	3	8	1	7	7	7
5	1	1	7	4	8	6
	2	4	2	3	7	4
	3	4	6	2	7	3
6	1	6	8	5	4	9
	2	10	6	1	4	9
	3	10	4	5	3	9
7	1	3	5	6	7	8
	2	4	3	4	7	8
	3	9	3	2	6	7
8	1	5	6	5	6	6
	2	6	5	1	6	5
	3	6	2	3	3	5
9	1	6	8	10	4	6
	2	7	5	8	4	4
	3	8	2	7	3	3
10	1	1	7	9	8	2
	2	10	7	3	6	1
	3	10	6	7	6	2
11	1	1	8	8	10	4
	2	2	7	7	9	3
	3	6	6	7	9	1
12	1	4	8	6	6	8
	2	5	7	3	3	6
	3	10	5	2	1	3
13	1	0	0	0	0	0
Resources maximum						
R1	R2	N1	N2			
15	18	76	70			

R: renewable; N: non-renewable.

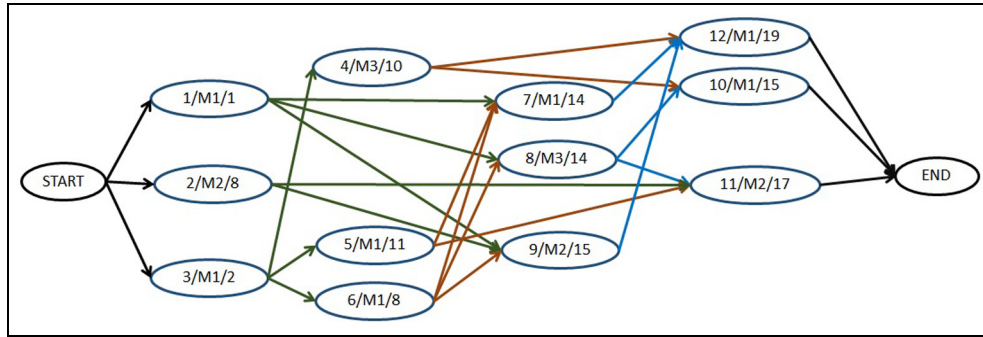


Figure 2. GGR-PSO solution for illustrative J12 instance.

Table 3. Various PSO combinations used for evaluation purposes.

Case	Activity priority			Activity mode
	ω, χ	Topology	GGR	
1	$\omega: 0.8$	gbest lbest rand-link group	Non-GGR/GGR curves (linear increasing curve/ Sugeno increasing ($s = -0.7$) curve/Sugeno increasing ($s = 10$) curve/S-increasing curve/dual S-increasing curve/sigmoid curve)	$\chi: 1$
2	$\chi: 0.72984$	gbest lbest rand-link group	Non-GGR/GGR curves (linear increasing curve/ Sugeno increasing ($s = -0.7$) curve/Sugeno increasing ($s = 10$) curve/S-increasing curve/dual S-increasing curve/sigmoid curve)	$\chi: 1$

PSO: particle swarm optimization; GGR: global guidance ratio.

to enable a fair comparison to be made with the solutions obtained using the existing schemes in the literature. As shown in Table 3, two GGR-PSO variants were applied in solving each MRCPS problem (Case 1 and Case 2). For both cases, the activity mode sub-problem was solved using the discrete PSO method. However, in Case 1, the activity priority sub-problem was solved using the conventional PSO scheme, while in Case 2, the activity priority sub-problem was solved using the standard PSO scheme. For both cases, the conventional/standard PSO schemes were implemented using the gbest, lbest, rand-link topology, and group topology. Finally, for both Case 1 and Case 2, the solution procedure was performed both without the GGR velocity update mechanism and with the GGR update mechanism based on six different linear and non-linear curve functions, respectively.

The performance of each PSO optimization scheme was evaluated initially in terms of the deviation ($Dev.BKS$) of the obtained solutions for each class of MRCPS from the best known solution (BKS), as defined in equation (14), in which $best_i$ is the BKS (makespan) for instance number i of the related class. The performance was further evaluated in terms of the percentage of instances for which the solution obtained by the PSO scheme was equal to the optimal solution (makespan) given in PSPLIB (see equation (15)). The

optimal makespan for J10 and J20 is reported in PSPLIB. However, for J30, optimal solutions are not yet available for all the instances. Thus, an additional comparison criterion “ $Incr.CP$ (%)” was derived with the form shown in equation (16), in which CP denotes the critical path (rather than optimal makespan) of the corresponding instance

$$Dev.BKS = \frac{\sum_{instances} \left(\frac{fitness_i - best_i}{best_i} \times 100\% \right)}{|instances|} \quad (14)$$

$$OPT = \frac{\sum_{instances} V_i}{|instances|} \times 100\% \quad (15)$$

$$Incr.CP = \frac{\sum_{instances} \left(\frac{fitness_i - CP_i}{CP_i} \times 100\% \right)}{|instances|} \quad (16)$$

Figures 3–5 show the OPT results of the two PSO schemes (Case 1 and Case 2) when applied to the J10, J20, and J30 MRCPSs, respectively. Note that symbols Non, L, Su $_{-0.7}$, Su $_{10}$, S, DS, and sigmoid denote non-GGR, linear increasing curve, Sugeno increasing ($s = -0.7$) curve, Sugeno increasing ($s = 10$) curve, S-increasing curve, dual S-increasing curve, and sigmoid curve, respectively. Note also that in Figure 3, the performance of the two PSO schemes for the J30

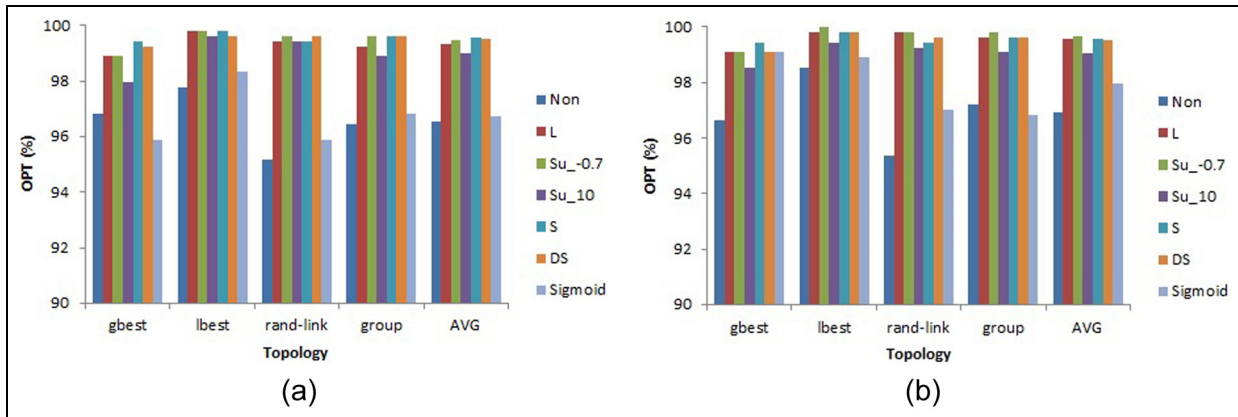


Figure 3. *OPT* comparison of 5000 feasible schedules for J10: (a) conventional PSO and (b) standard PSO.

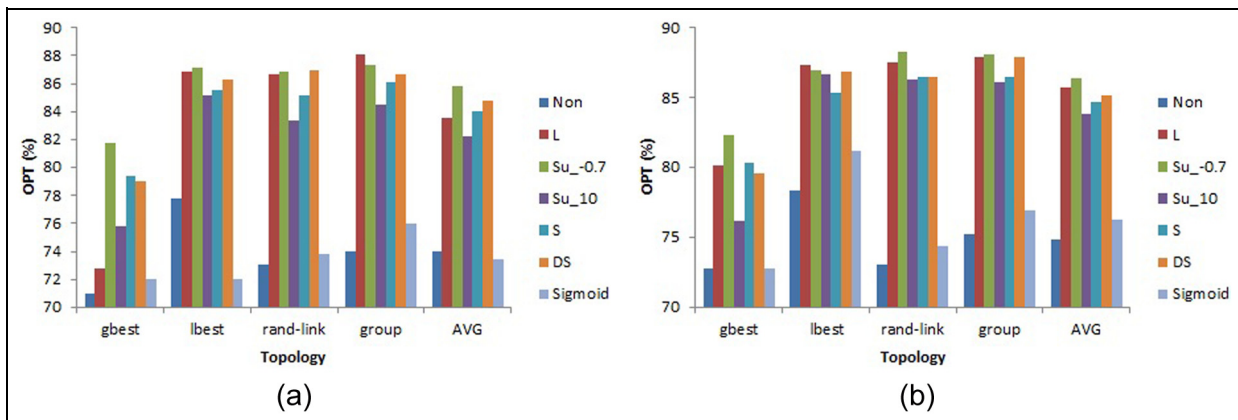


Figure 4. *OPT* comparison of 5000 feasible schedules for J20: (a) conventional PSO and (b) standard PSO.

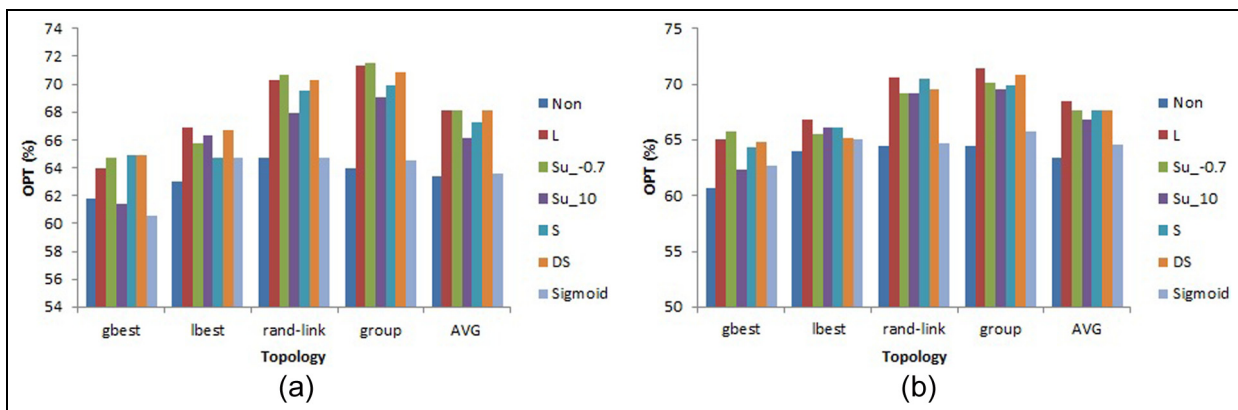


Figure 5. *OPT* comparison of 5000 feasible schedules for J30: (a) conventional PSO and (b) standard PSO.

instances is evaluated by comparing the PSO solution with the “best” rather than “optimal” reported solution in the literature. In general, the results show that the use of the GGR mechanism yields a better *OPT* performance than that obtained when the GGR mechanism

is not applied. Furthermore, among the six curve functions used to compute the GGR value, the sigmoid curve yields the poorest PSO solution. (It is noted, however, that the solutions are still better than those obtained when the GGR mechanism is not applied.)

Table 4. Average *OPT* performance of different algorithms for J10, J20, and J30 using conventional PSO scheme.

Topology	Non	L	Su _{-0.7}	Su ₁₀	S	DS	Sigmoid
gbest	76.51	78.52	81.77	78.39	81.24	81.06	76.14
lbest	79.53	84.49	84.25	83.71	83.35	84.19	78.34
rand-link	77.64	85.46	85.70	83.59	84.73	85.64	78.13
group	78.14	86.24	86.18	84.13	85.22	85.70	79.10

Table 5. Average *OPT* performance of different algorithms for J10, J20, and J30 using standard PSO scheme.

	Non	L	Su _{-0.7}	Su ₁₀	S	DS	Sigmoid
gbest	76.75	81.42	82.38	79.00	81.36	81.18	78.16
lbest	80.27	84.68	84.19	84.07	83.77	83.95	81.71
rand-link	77.64	86.00	85.76	84.91	85.46	85.22	78.69
group	78.99	86.30	86.00	84.91	85.34	86.30	79.83

Comparing the remaining curve functions, the L, Su_{-0.7}, S, and DS curves yield a better solution performance than the Su₁₀ and Sigmoid curves. Finally, for the cases with fewer activities (i.e. J10 and J20), the lbest topology results in the best solution performance. However, for a greater number of activities (i.e. J30), the group topology results in an improved solution performance.

Tables 4 and 5 summarize the average *OPT* results obtained for cases J10, J20, and J30 when using the conventional PSO and standard PSO schemes, respectively, with the four different communication topologies and six different GGR curve functions. For the standard PSO algorithm, the L or DS curves with the group topology result in the best overall performance of the various schemes over all the considered instances, that is, an average *OPT* value of 86.30%. However, the *OPT* performance obtained using the L curve is better than that obtained using the DS curve for all the other topologies. Thus, in comparing the performance of the proposed GGR-PSO scheme with that of other state-of-the-art MRCPS algorithms presented in the literature, the activity operating priority problem was solved using the group topology mechanism and the linear increasing GGR update curve.

Tables 6 and 7 compare the performance of the proposed GGR-PSO scheme (group topology and linear increasing curve) for cases J10–J20 and J30, respectively, with that of other methods presented in the literature. (Note that the activity operating priority problem was solved using the standard PSO method.) The performance is evaluated over 5000 feasible solutions for each instance. For cases J10–J20, the average *OPT* of the proposed method (J10: 99.63%, J20: 87.91) is close to that of the best performing scheme (Chiang et al.,² i.e. J10: 99.81%, J20: 88.27%). In addition, the

GGR-PSO method achieves the lowest *Dev.BKS* among all the considered schemes. For case J30, the average *Dev.BKS* performance of the proposed method (1.19%) is close to that of the best performing scheme (Van Peteghem and Vanhoucke;²² i.e. 1.08%), while the average *equal* value of the proposed method (71.38%) is superior to that of Van Peteghem et al. (71.0%). In addition, the average *equal* performance of the proposed method is also close to that of the best performing scheme (Chiang et al.;² i.e. 72.64%). The average *Dev.BKS* of the proposed method (1.19%) is less than that of Chiang et al. (2.6%).

The search performance of the proposed scheme relative to that of the existing methods can be evaluated using the metric shown in equation (17). Table 8 indicates the corresponding results. In average, this study is able to obtain the largest optimal solutions. As shown, the search performance of the proposed scheme is around 8.7% better than that of the existing algorithms when evaluated over cases J10–J20, and 12.91% better when evaluated over cases J10–J30

$$\text{Improvement}(\%) = \frac{\text{Avg.OPT}(\text{this work}) - \text{Avg.OPT}(\text{algorithm})}{\text{Avg.OPT}(\text{algorithm})} \times 100\% \quad (17)$$

In evaluating the complexity of the proposed method, consider an MRCPS instance consisting of N activities, where each activity has M possible execution modes. The solution space for the MRCPS instance therefore contains $M^N \times N!$ possible solutions. In other words, the computational complexity is $O(M^N \times N!)$. When simulating a median scale MRCPS instance such as J18 ($N = 18$, $M = 3$), the time required to perform an exhaustive search is

Table 6. Dev.BKS and OPT performance of different algorithms for cases J10–J20 (5000 schedules).

		Instance set						
		J10	J12	J14	J16	J18	J20	Avg.
Dev.BKS (%)								
This work		0.02	0.05	0.25	0.20	0.36	0.46	0.17
Van Peteghem and Vanhoucke ²²		0.01	0.09	0.22	0.32	0.42	0.57	0.27
Coelho and Mario ²³		0.07	0.16	0.32	0.48	0.56	0.80	0.40
Wang and Chen ²⁴		0.10	0.21	0.46	0.57	0.94	1.40	0.61
Chiang et al. ²		0.34	N/A	N/A	N/A	N/A	1.79	1.06
Wang and Chen ²⁵		0.12	0.14	0.43	0.59	0.90	1.28	0.58
Li and Hong ²⁶		0.09	0.13	0.40	0.57	1.02	1.10	0.55
Chen and Wu ¹⁴		0.02	0.09	0.38	0.46	0.69	0.89	0.42
OPT (%)								
This work		99.63	98.72	94.19	94.73	90.58	87.91	
Van Peteghem and Vanhoucke ²²		99.63	98.17	94.56	92.00	88.95	85.74	
Wang and Chen ²⁴		97.93	95.97	90.86	86.49	79.44	72.84	
Chiang et al. ²		99.81	N/A	N/A	N/A	N/A	88.27	
Li and Hong ²⁶		98.30	96.50	90.30	86.90	76.10	72.40	
Chen and Wu ¹⁴		99.63	98.53	91.11	89.45	84.42	81.04	

N/A: not available.

Table 7. Performance comparison of different algorithms for J30.

	Dev.BKS (%)	Incr.CP (%)	Worse (%)	Equal (%)
This work	1.19	13.72	28.62	71.38
Van Peteghem and Vanhoucke ²²	1.08	13.75	29.00	71.00
Wang and Chen ²⁴	N/A	13.46	N/A	N/A
Chiang et al. ²	2.60	N/A	27.36	72.64
Li and Hong ²⁶	5.56	N/A	56.20	43.80
Chen and Wu ¹⁴	1.79	14.81	30.62	69.38

N/A: not available.

Table 8. Search performance improvement of proposed scheme compared to existing algorithms.

	Avg.OPT (%)		Improvement (%)	
	J10:J20	J10:J30	J10:J20	J10:J30
This work	94.29	91.02		
Van Peteghem and Vanhoucke ²²	93.18	90.01	1.20	1.13
Wang and Chen ²⁴	87.26	N/A	8.07	N/A
Chiang et al. ²	94.04	86.91	0.27	4.73
Li and Hong ²⁶	86.75	80.61	8.70	12.91
Chen and Wu ¹⁴	90.70	87.65	3.97	3.84

N/A: not available.

equal to $2.5 \times 10^{24} \times 10^{-8}$ s, that is, 2.9×10^{11} days (assuming a solution time of $0.01 \mu\text{s}$ (10^{-8} s) for each solution). In other words, solving the MRCPSP problem is extremely time-consuming. However, in this study, the solution procedure for each instance is terminated once 5000 feasible schedules have been obtained. The central processing unit (CPU) time

required to obtain 5000 schedules for a typical J18 instance is approximately 0.74 s. According to Table 6, the proposed scheme finds more than 90% of the optimal solutions for the J18 case in 410 s. In other words, the PSO metaheuristic algorithm proposed in this study greatly reduces the computational complexity compared to that of exhaustive methods.

Overall, the results presented in Tables 6–8 show that the proposed GGR-PSO scheme based on the global topology communication paradigm and a linear increasing GGR function provides a highly effective and competitive method for solving MRCPSPs.

Conclusion

This study has proposed a PSO-based scheme designated as GGR-PSO for solving MRCPSPs. In the proposed method, the activity priority sub-problem is solved using the conventional or standard PSO scheme, while the activity operating mode sub-problem is solved using the discrete PSO method. Notably, to enhance the efficiency in solving the activity priority sub-problem, an adaptive GGR parameter is used to weight the effects of the local experience of the particles and the global experience of the swarm on the updated particle velocity in each iteration round. Through the use of the GGR parameter, the particles explore a diverse range of candidate solutions in the earlier stages of the solution search procedure, but converge toward the globally optimal solution in the later stages. Consequently, the robustness of the search procedure is improved. The performance of the proposed GGR-PSO scheme has been evaluated for four different communication topologies (gbest, lbest, rand-link, and group) and six different GGR computation schemes (linear increasing curve, Sugeno increasing ($s = -0.7$) curve, Sugeno increasing ($s = 10$) curve, S-increasing curve, dual S-increasing curve, and sigmoid curve). Moreover, the group topology used to determine the global experience to balance the convergence of global topology and local topology is also suggested. The experimental results have shown that the optimal performance of the proposed GGR-PSO scheme is obtained using a group topology and the linear increasing curve. The validity of the proposed scheme has been demonstrated by solving the MRCPSP cases presented in PSPLIB. It has been shown that the scheduling solutions obtained using the algorithm proposed by Chiang et al.² are in slightly better agreement with the optimal solutions (J10 and J20) and best solutions (J30) in PSPLIB than those obtained using the proposed GGR-PSO scheme. However, for all the cases, the solutions obtained using the GGR-PSO scheme deviate less significantly from the BKS. Thus, it is inferred that the proposed GGR-PSO method provides an effective and robust technique for solving MRCPSPs.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was partly supported by the Ministry of Science and Technology, Taiwan, under contract MOST 104-2221-E-167-011.

References

1. Blazewicz J, Lenstra JK and Rinnooy Kan AHG. Scheduling subject to resource constraints: classification and complexity. *Discrete Appl Math* 1983; 5: 11–24.
2. Chiang CW, Huang YQ and Wang WY. Ant colony optimization with parameter adaptation for multi-mode resource-constrained project scheduling. *J Intell Fuzzy Syst* 2008; 19: 345–358.
3. Ranjbar M, De Reyck B and Kianfar F. A hybrid scatter search for the discrete time/resource trade-off problem in project scheduling. *Eur J Oper Res* 2008; 193: 35–48.
4. Magalhães-Mendes J. A two-level genetic algorithm for the multi-mode resource-constrained project scheduling problem. *Int J Syst Appl Eng Dev* 2011; 3: 271–278.
5. Jarboui B, Damak N, Siarry P, et al. A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems. *Appl Math Comput* 2008; 195: 299–308.
6. Chen RM. Particle swarm optimization with justification and designed mechanisms for resource-constrained project scheduling problem. *Expert Syst Appl* 2011; 38: 7102–7111.
7. Ziarati K, Akbari R and Zeighami V. On the performance of bee algorithms for resource-constrained project scheduling problem. *Appl Soft Comput* 2011; 11: 3720–3733.
8. Bouleimen K and Lecocq H. A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version. *Eur J Oper Res* 2003; 149: 268–281.
9. Damak N, Jarboui B, Siarry P, et al. Differential evolution for solving multi-mode resource-constrained project scheduling problems. *Comput Oper Res* 2009; 36: 2653–2659.
10. Marinakis Y and Marinaki M. A hybrid multi-swarm particle swarm optimization algorithm for the probabilistic traveling salesman problem. *Comput Oper Res* 2010; 37: 432–442.
11. Chen RM and Shen YM. Novel encoding and routing balance insertion based particle swarm optimization with application to optimal CVRP depot location determination. *Math Probl Eng* 2015; 2015: 743507.
12. Prins C. Two memetic algorithms for heterogeneous fleet vehicle routing problems. *Eng Appl Artif Intel* 2009; 22: 916–928.
13. Chen W, Shi YJ, Teng HF, et al. An efficient hybrid algorithm for resource-constrained project scheduling. *Inform Sciences* 2010; 180: 1031–1039.
14. Chen RM and Wu DS. Solving scheduling problem using particle swarm optimization with novel curve based inertia weight and grouped communication topology. *Int J Digit Content Technol Appl* 2013; 7: 94–103.

15. Kolisch R and Sprecher A. PSPLIB—a Project Scheduling Problem Library: OR Software—ORSEP Operations Research Software Exchange Program. *Eur J Oper Res* 1997; 96: 205–216.
16. Kennedy J and Eberhart RC. Particle swarm optimization. In: *Proceedings of the IEEE international conference on neural networks*, Perth, WA, Australia, 27 November–1 December 1995, vol. 4, pp.1942–1948. New York: IEEE.
17. Bratton D and Kennedy J. Defining a standard for particle swarm optimization. In: *Proceedings of the IEEE swarm intelligence symposium (SIS '07)*, Honolulu, HI, 1–5 April 2007, pp.120–127. New York: IEEE.
18. Kennedy J and Eberhart RC. A discrete binary version of the particle swarm algorithm. In: *Proceedings of the IEEE conference on systems, man, and cybernetics*, Orlando, FL, 12–15 October 1997, vol. 5, pp.4104–4109. Piscataway, NJ: IEEE.
19. Lei K, Qiu Y and He Y. A new adaptive well-chosen inertia weight strategy to automatically harmonize global and local search ability in particle swarm optimization. In: *Proceedings of the 1st international symposium on systems and control in aerospace and astronautics (ISSCAA)*, Harbin, China, 19–21 January 2006, pp.977–980. New York: IEEE.
20. Lee KC and Ou JS. Adaptive color image enhancement using particle swarm optimization. *J Inform Electron* 2007; 2: 29–38.
21. Malik RF, Rahman TA, Hashim SZ, et al. New particle swarm optimizer with sigmoid increasing inertia. *Int J Comput Sci Secur* 2007; 1: 35–44.
22. Van Peteghem V and Vanhoucke M. A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem. *Eur J Oper Res* 2010; 201: 409–418.
23. Coelho J and Mario V. Multi-mode resource-constrained project scheduling using RCPSP and SAT solvers. *Eur J Oper Res* 2011; 213: 73–82.
24. Wang L and Chen F. An effective shuffled frog-leaping algorithm for multi-mode resource-constrained project scheduling problem. *Inform Sciences* 2011; 181: 4804–4822.
25. Wang L and Chen F. An effective estimation of distribution algorithm for the multi-mode resource-constrained project scheduling problem. *Comput Oper Res* 2012; 39: 449–460.
26. Li H and Hong Z. Ant colony optimization-based multi-mode scheduling under renewable and nonrenewable resource constraints. *Automat Constr* 2013; 35: 431–438.