Iraqi Journal of Science

ISSN: 0067-2904

Using Non-dominated Sorting Particle Swarm Optimization Algorithm II for Bi-objective Flow Shop Scheduling Problems

Hanan Ali Chachan*, Faez Hassan ali
Department of Mathematics, College of Sciences, Mustansireyah University, Baghdad, Iraq

Received: 15/8/2020 Accepted: 17/10/2020

Abstract

A hybrid particulate swarm optimization (hybrid) combination of an optimization algorithm of the particle swarm and a variable neighborhood search algorithm is proposed for the multi-objective permutation flow shop scheduling problem (PFSP) with the smallest cumulative completion time and the smallest total flow time. Algorithm for hybrid particulate swarm optimization (HPSO) is applied to maintain a fair combination of centralized search with decentralized search. The Nawaz-Enscore-Ham (NEH) heuristic algorithm in this hybrid algorithm is used to initialize populations in order to improve the efficiency of the initial solution. The method design is based on ascending order (ranked-order-value, ROV), applying the continuous PSO algorithm to the PFSP, introducing the external archive set storage Pareto solution, and using a hybrid strategy that combines strong dominance and aggregation distance to ensure the distribution of the solution set. We adopted the Sigma method and the roulette method, based on the aggregation distance, to select the global optimal solution. A variable neighborhood search algorithm was proposed to further search the Pareto solution in the external set. The suggested hybrid algorithm was used to solve the Taillard test set and equate the test results with the SPEA2 algorithm to check the scheduling algorithm's efficacy.

Keywords: particle swarm optimization algorithm; variable neighborhood search; multi-objective; permutation flow shop scheduling.

الخوارزمية الثانية لمشاكل جدولة متجر التدفق استخدام تحسين سرب الجسيمات الفرز غير المسيطر ثنائي الهدف

حنان علي جيجان، فائز حسن علي
قسم الرياضيات، كلية العلوم، الجامعة المستنصريه، بغداد، العراق

الخلاصه

تم اقتراح مجموعة تحسين (هجين) لسرب الجسيمات الهجين من خوارزمية التحسين لسرب الجسيمات وخوارزمية بحث متغيرة المجاورة لمشكلة جدولة متجر التدفق متعدد الأهداف (PFSP) مع أصغر وقت إتمام تراكمي وأصغر وقت تدفق إجمالي. خوارزمية لتحسين سرب الجسيمات (HPSO) وللحفاظ على مزيج عادل من البحث المركزي مع البحث اللامركزي. تُستخدم خوارزمية (Nawaz–Enscore–Ham) NEH التجريبية في هذه الخوارزمية الهجينة لتهيئة السكان من أجل تحسين كفاءة الحل الأولي. يعتمد تصميم الطريقة على

*Email: Hananaltaai1970@gmail.com

ترتيب تصاعدي (قيمة ترتيب مرتبة ، ROV)، مع تطبيق خوارزمية PSO المستمرة على مشكلة جدولة متجر التدفق البديل ؛ تقديم حل Pareto لتخزين مجموعة الأرشيف الخارجي ، واستخدام إستراتيجية مختلطة تجمع بين الهيمنة القوية ومسافة التجميع.ضمان توزيع مجموعة الحلول ؛ اعتماد طريقة سيجما وطريقة الروليت على أساس مسافة التجميع لاختيار الحل الأمثل العالمي ؛ اقتراح خوارزمية بحث متغيرة في الجوار لمزيد من البحث عن حل باريتو في المجموعة الخارجية. استخدام الخوارزمية المختلطة المقترحة لحل مجموعة اختبار Taillard ومساواة نتائج الاختبار بخوارزمية. SPEA2 للتحقق من فعالية خوارزمية الجدولة

Introduction

The problem of scheduling the replacement flow shop is probably the most commonly discussed example of traditional scheduling problems, and work on this subject is of great significance. Throughout the past few decades, researchers conducted a remarkable amount of work in this field, most of which concentrating on solving the single-objective PFSP. Nevertheless, they also faced the issue of multi-objective decision-making in the real production phase. Thus, learning the algorithm to solve the multi-objective PFSP is of considerable theoretical importance and real interest in electronics.[1]

Research on multi-objective scheduling, based on smart algorithms, has slowly gained comprehensive attention from academia and engineering fields in recent years. In fact, well-known journals such as the "European Operations Analysis" have successively published accompanying special issues, supporting multi-objective intelligent algorithms in the laboratory too as well as applications on the problem of scheduling.[2]

Alaa and Hanab [3] merged genetic algorithms and local search techniques and used random-based adaptive weighted functions to determine solutions and direct genetic operations. [4,5,6,7] They also introduced a multi-objective local genetic search (MOGLS) algorithm. Likewise, Hanan [7] used a random weight-based adaptive value function, selected the parent person and local search guides, and then provided a multi-objective genetic hybrid algorithm. Furthermore, Abbas. [8] proposed a type of virtual multi-objective annealing algorithm, which generates a set of random weight vectors to construct a set of evaluation functions and then performs a local search based on each evaluation function under the guidance of a simulated annealing strategy.

In comparison to the above methods, Zilter E.[9] and Mostaaghim S [10] they followed the idea of Pareto dominance which identifies the existing population and assigns correct adaptation values to the population based on this. While, at the same time, a parallel multi-target local population search is performed to improve the search capacity in areas with low solution density in the target region. Farid and Wang[11 ,12] have recently developed a hybrid genetic algorithm based on quantum computation, which also uses a fast sorting and modification value assignment assessment process, where the algorithm was implemented to eliminate the redundancy of human populations.

Particle Swarm Optimization (PSO) algorithm is a modern type of group-based optimization algorithm that Taillard E. [13] first proposed in 1995. The PSO algorithm is generated by the algorithm of optimization of Swarm Intelligence by cooperation and competition between particles within the population. The PSO algorithm maintains the population-based global search approach relative to the genetic algorithm. The search paradigm is simple and has the characteristics of quick convergence and high robustness. The early PSO algorithm was used to refine unconstrained continuous functions and was widely extended to other problems, such as voltage regulation and training on neural network. Jawad and Ali. [14] successfully used the PSO algorithm to solve the problem of single-objective substitution flow shop scheduling, supporting the PSO algorithm in a separate implementation of the Combinatorial Optimization Problem.

However, the latest work on multi-objective scheduling algorithms for particle swarm optimization is still very small. Recently suggested by Carles et al.. [15], the descriptive function is a multi-objective PSO algorithm that is used after the time is weighted to handle the evaluation link.

A related works[16,17] introduced a modal analysis of vibration of Euler–Bernoulli beam subjected to concentrated moving load. In another studys [18,19,20,21,22], the same flow was also presented on the field of optimization problem using heuristic and branch and bound methods to solve a multi-criteria machine scheduling problem. In addition, a genetic algorithm-based anisotropic diffusion filter and clustering algorithms for thyroid tumor detection were introduced to solve some problems in the same area [23,24].

Based on existing research, a hybrid particle swarm optimization algorithm is designed in this paper, combining the advantages of the particle swarm optimization algorithm and the variable neighborhood search algorithm, to solve the dual-objective flow shop scheduling problem. The hybrid algorithm uses ascending order the rule (ROV) continuous PSO algorithm conducts a global search. Then, it uses a critical path-based vector neighborhood search algorithm to conduct a local search on globally optimized particles, such that the algorithm maintains a fair compromise between localized search and centralized search. Based on the concept of Pareto domination, an effective non-inferior solution set update strategy was designed. Simulation experiments were conducted on the benchmark test set, proposed by Taillard, and compared with other multi-objective algorithms to confirm the effectiveness of the algorithm.

Problem Description

Description of multi-objective optimization problem

$$\min f(x) = (f_1(x), f_2(x), \dots, f_N(x)) \quad x \in X \tag{1}$$

where f_1, f_2, …, f_N are N objective functions, x represents the decision solution, and X represents the solution space.

Consider two solutions x_1 and x_2. If

$$\begin{cases} \forall j \in \{1,2,\cdots,N\}, f_j(x_1) \leq f_j(x_2) \\ \exists k \in \{1,2,\cdots,N\}, f_k(x_1) \leq f_k(x_2) \end{cases} \tag{2}$$

then the solution x_1 dominates the solution x_2, denoted as $x_1 \succ x_2$.

Given a solution x^* (optimal solution), if there is no solution that dominates $x *$ in the solution space X, then x^* is called the Pareto optimal solution, or a non-dominated solution. The set of all fee-dominated solutions constitutes the optimal solution set in the multi-objective sense. these solutions constitute the Pareto Front of the problem in the objective space.

Description of permutation flow shop scheduling problem

The scheduling of the replacement flow shop can be described as [8]: n workpieces are to be processed on m machines. The processing sequence of each workpiece is the same, the sequence of workpieces processed by each machine is also the same, and the processing time of each workpiece on each machine. Scheduling objectives generally includes: minimizing the maximum workpiece delay; minimizing the total workpiece flow time (TFT); maximizing the completion time of all workpieces (Makespan) being the shortest; average workpiece waiting time being the shortest, etc. [8]. In this paper, maximum completion time and total flow time are the shortest of the two goals. The following assumptions are usually made on this problem:

1) A workpiece can only be processed on one machine at a time;
2) One machine can only process one workpiece at a time;
3) Once the workpiece is processed on a certain machine, it cannot be stopped;
4) The machining sequence of the workpiece on each machine is the same.

The mathematical description of the scheduling problem with replacement flow shop is as follows. Let t ijbe the processing time of the workpiece I on machine j, except the preparation period needed for processing the workpiece j immediately after processing the workpiece I on the same machine. C(π i,j) is the workpiece. The completion period of π i in machine j, without lack of generality, assumes that each workpiece is processed in the order of machine 1 to m, let $\pi = \{\pi_1, \pi_2, \dots, \pi_n\}$ be a sort of all workpieces.

$$\begin{cases} C(\pi_1,1) = t_{\pi_1,1} \\ C(\pi_i,1) = C(\pi_{i-1},1) + t_{\pi_1,1}, i = 2,\cdots,n \\ C(\pi_i,j) = C(\pi_1,j-1) + t_{\pi_1,j}, j = 2,\cdots,m \\ C(\pi_i,j) = \max\{C(\pi_{i-1},j), C(\pi_i,j-1)\} + t_{\pi_i,j} \\ \qquad\qquad i = 2,\cdots,n \quad j = 2,\cdots,m \end{cases} \tag{3}$$

$$makespan = C_{\max}(\pi) = C(\pi_n, m) \quad (4)$$

$$TFT = \sum_{i=1}^{n} F(\pi_i) = \sum_{i=1}^{n} C(\pi_i, m) \quad (5)$$

Equations (4) and (5) are the calculation formulas for maximum completion time and total flow time, respectively.

Hybrid particle swarm optimization algorithm for solving multi-objective replacement flow shop scheduling problem

Standard Particle Swarm Optimization (PSO)

The optimization algorithm for base particle swarm uses a speed-position model to search. Each candidate solution that needs to be optimized is called a "particle". Each particle has its own position and speed, and the optimization function determines an adaptation value. Every particle memorizes, follows the best current particle, and looks for space inside the solution. Every iteration process is not entirely random. Based on this, the next solution should be sought until a better solution is discovered. The PSO algorithm is initialized into random particle category A. In each iteration, the particle updates itself by measuring two "extreme values": the first is the best solution found by the particle itself, called the specific extreme point (using p_{best} to represent its position), globally. In the PSO version, the other extreme point is the best solution found by the whole population, called the global extreme point (the best is used to represent its position). Local extreme point (using l_{best} to reflect position) is the best answer. After finding these two best solutions, the particle updates its own velocity and position according to the following equations (6 and 7). The information of particle i can be expressed by n-dimensional vector, the position is expressed as$X_i = [x_{i1}, x_{i2}, \ldots, x_{in}]$, the speed is$V_i = [v_{i1}, v_{i2}, \ldots, v_{in}]$, and other vectors are similar. Then the speed and position updated equations are:

$$v_{ij}^{t+1} = w \times v_{ij}^{t} + c_1 r_1 \left(P_j^t - X_{ij}^t\right) + c_1 r_1 \left(G_j^t - X_{ij}^t\right) \quad (6)$$

$$X_{ij}^{t+1} = v_{ij}^{t+1} + X_{ij}^{t} \quad (7)$$

where $w(t) = w(t-1)$ spectrum is the inertia coefficient, β is the linear decreasing component and its key function is to produce disturbances to avoid premature convergence of the algorithm, c1 and c2 are learning variables that are tuned to the best individual particles and the most global the maximum phase size for flying in a successful particle direction. If too small, the particles may be far removed from the target area. At the same time, it is not easy to fall into the optimum locale; typically $c1 = c2 = 2$; $r1$ and $r2$ are uniformly generated random numbers between [0, 1].

Particle swarm optimization algorithm for PFSP

Representation of solutions and ROV rules

The most widely used way of coding for the PFSP problem in the literature is to use the ordering of the workpieces explicitly. Since the position of the particles in the continuous PSO algorithm is a continuous value vector to achieve the mapping relationship between the particle position vector and the order of the workpieces, Zitler and Thiele. [9] suggested the ROV rule to transform the continuous position vector of the particle $X\,i = [x\,i1, x\,i2, \ldots, x\,in]$ into a discrete order of processing.

The ROV rule is laid down as follows. First, assigning the ROV value of one to the component position with the smallest value for a particle position vector, then assigning the ROV value of 2 to the second smallest component position, and so on, until all component positions are assigned a unique ROV value, so that a workpiece order can be constructed based on the ROV value.

In Table 1, a basic example is used to demonstrate the ROV rule construction procedure. Consider the replacement flow shop scheduling of 7 workpieces, the position vector of the particle is 6-dimensional, and set the position vector X i= [0.46, 2.70, 1.53, 0.81, 2.61, 0.25, and 1.08]. It can be seen that xi6 is the smallest, so that the ROV value of the component position corresponding to xi6 is assigned to 1, and then the ROV value of the component position corresponding to xi1 is assigned to 2. The ROV values of the component positions corresponding to xi5 and xi2 are 3, 4, 5, 6, and 7, so as to obtain the processing order of the workpiece, that is, π={2, 7, 5, 3, 6, 1, 4}.

Table 1-ROV value corresponding to particle position vector

Position	1	2	3	4	5	6	7
Position value	0.46	2.70	1.53	0.81	2.61	0.25	1.08
ROV value	2	7	5	3	6	1	4

Population initialization

The initial population will have any distribution, which with a great likelihood will occupy the whole solution space. However, in order to increase community search performance and prevent blind search, some of the better performing approaches would also be included in the initial sample. Therefore, the initial solution is generated. There are essentially two methods, one is generating spontaneously in a continuous interval and the other is generating by using a heuristic method of construction. Here, we use the highest performing NEH heuristic algorithm [23,24], which includes the following algorithm measures proposed jointly by Nawaz, Enscore, and Ham:

Step 1: Arrange the n workpieces in order to decrease total processing time of the workpieces on the machine.

Step 2: Schedule the first two parts to minimize the total completion time.

Step 3: From k=3, …, n, insert the k-th work piece into k possible positions, and find the smallest part of the total completion time.

The solution generated by the NEH heuristic is a sequence of workpieces, which must be converted into a position vector within a certain interval. Here, the conversion is implemented as follows.

$$x_{NEH,j} = x_{min,j} + \frac{x_{max,j} - x_{min,j}}{n} \times \left(s_{NEH,j} - 1 + r\right) \quad j = 1,2,\dots,n \qquad (8)$$

where $x_{NEH,j}$ is the position value of the particle in the jth dimension, $s_{NEH,j}$ is the number of the jth dimension workpiece obtained by the NEH method, $x_{max,j}$ and $x_{min,j}$ are the upper bounds of the particle position vector on the continuous space, respectively, and the lower bound, $r3$, represents a randomly generated random number between 0 and 1.

The random generation method is: $X_{ij}^{0} = X_{min} + (X_{max} - X_{min}) \times r$ $V_{ij}^{0} = V_{min} + (V_{max} - V_{min}) \times r$ where the position vector value changes between continuous intervals$[X_{min}, X_{max}]$. The velocity vector value changes in a continuous$[V_{min}, V_{max}]$, while r1 and r2 are random numbers generated uniformly between 0 and 1.

Multi-objective evolution strategy

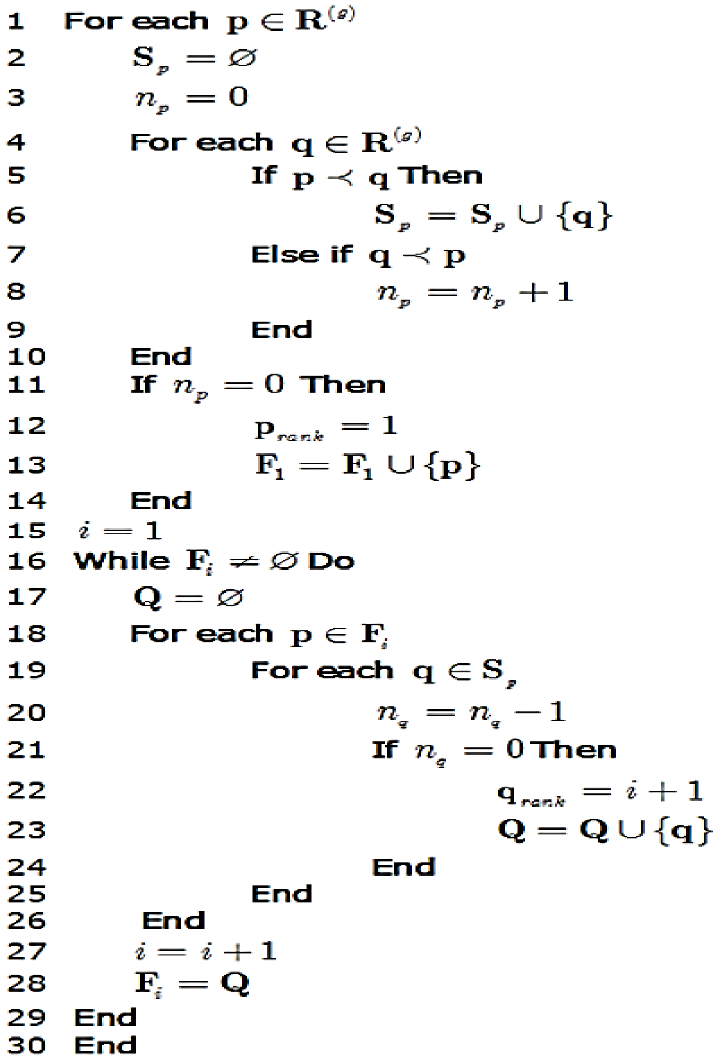
The multi-objective evolutionary algorithm needs to ensure that Pareto's frontier is characterized by convergence and diversity. The key is to set a reasonable Pareto strategy for the maintenance of diversity and an optimal value update operation for the global particle swarm. In this article, using an external set (ES) for storage, a rapid Pareto-based sorting approach is used to construct non-dominated collection, incorporating an archive technique that incorporates a clear dominance relationship and aggregation distance to allow the Pareto frontier to have reasonable distribution. Two phases are also followed in the evaluation of the desired global solution; Management approach, using the Sigma method for fast convergence in the early stage, and the roulette method based on the aggregation distance for later in-depth search.

Generation of external collection ES

After the initial population is generated, the target value of each particle is determined. The quick sorting method is used to sort the particles according to the target function value and the non-dominated solution is stored in the external set ES.

The principle of fast sort is that each loop selects a person x from the population (usually the first one), and in turn compares the other individuals in the population with x. A contrast splits the species into two parts; the second half of the population part is the organism occupied by x, while the first half is occupied by non-x-related individuals of x. If x is not dominated by any other individual, then x is incorporated into the non-dominated set, and then the above process is repeated for the first half until the first half is empty. The pseudocode for quick sorting is shown in Figure-1.

After the update of each generation of particles is completed, the fitness value of a single particle on each target is calculated, along with the new external collection ES. As shown in Figure 2, the update principle involves: (1) If the particle dominates some of the external collection of ES Particles, delete the dominated particles, and add the particles to the external set ES; (2) If there are particles in the external set ES that dominate the particles, they are ignored; (3) If the particles and the individual particles in the external set ES are mutually exclusive dominate, the particle is added to the external collection ES.



```
For each $\mathbf{p} \in \mathbf{R}^{(g)}$
    $\mathbf{S}_p = \varnothing$
    $n_p = 0$
    For each $\mathbf{q} \in \mathbf{R}^{(g)}$
        If $\mathbf{p} \prec \mathbf{q}$ Then
            $\mathbf{S}_p = \mathbf{S}_p \cup \{\mathbf{q}\}$
        Else if $\mathbf{q} \prec \mathbf{p}$
            $n_p = n_p + 1$
        End
    End
    If $n_p = 0$ Then
        $\mathbf{p}_{rank} = 1$
        $\mathbf{F}_1 = \mathbf{F}_1 \cup \{\mathbf{p}\}$
    End
$i = 1$
While $\mathbf{F}_i \neq \varnothing$ Do
    $\mathbf{Q} = \varnothing$
    For each $\mathbf{p} \in \mathbf{F}_i$
        For each $\mathbf{q} \in \mathbf{S}_p$
            $n_q = n_q - 1$
            If $n_q = 0$ Then
                $\mathbf{q}_{rank} = i + 1$
                $\mathbf{Q} = \mathbf{Q} \cup \{\mathbf{q}\}$
            End
        End
    End
    $i = i + 1$
    $\mathbf{F}_i = \mathbf{Q}$
End
End
```

Figure 1-Pseudo code for quick sort

Maintenance of external collection ES

When all the Pareto optimal solutions that exist in the evolution cycle are kept for complex multi-objective optimization problems, a significant amount of identical solutions may emerge in the external range, which does not only impact the solution's distribution efficiency but also decreases the solution's overhead memory and time for completeness. This paper adopts an archiving strategy to maintain the distribution of external collections, which combines strong dominance and aggregation distance to maintain external collections.

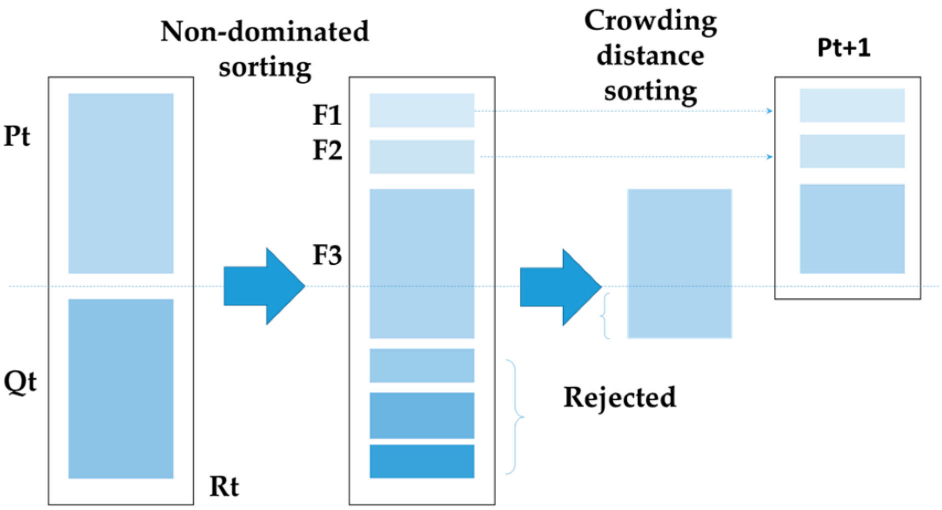


Figure 2-Principle of external collection update

The implementation of a strong dominance relationship allows each particle to maintain a certain distance, essentially regulating the degree of particle aggregation, and helping to sustain the particle group diversity. The concept of a close commitment to supremacy is:

1) If the particle $A \succ B$, then A f is considered strong B;

2) For A~B, ∀ objective function m, if $|f_m^A - f_m^B| < \varepsilon_m$, $(m = 1, 2, \ldots, k)$ is a normal number, then randomly choose A or B; if A is If selected, A f is considered to be strong B, and vice versa.

If ε_m is too low, there would be no impact of heavy supremacy. If ε_m=0, the strong dominance degenerates into a dominance; if the value is too large, the strong dominance becomes stronger, making the particles too sparse. It can be shown that it is very important to value γ m. For this paper, the mathematical formula is used to determine the value.

we have only two objectives for the case:

$$\varepsilon_m = \frac{f_m^{max} - f_m^{min}}{c}$$

C is the strong dominating coefficient, C>80.

The aggregation interval is an operator introduced in 2002 by Wang and Farid.[12] and used in NSGA-II for preserving population diversity. It can be used to describe the degree of solution density. The size of individuals aggregating with a low aggregation rate is very large. On each sub-target, the aggregation distance can be obtained by the sum of the distance differences between two adjacent individuals. The pseudo code for computing the distance between aggregations is as follows:

Selection of global optimal value

Since the global optimal solution has a very obvious guiding effect on the particles, how to choose a better global optimal solution g_{best} to guide the particle flight has a very important effect on the algorithm, which is related to the algorithm's convergence speed and solution diversity. We are following a two-stage leadership approach. Early stage uses the Sigma method for incredibly rapid convergence speed, and later stage uses the roulette method based on aggregation distance to perform a more in-depth scan.

Wang and Farid. [12] suggested a system of choosing leadership based on the Sigma standard. The basic idea is to assign a Sigma value for each particle in the community and an external file will determine the particle's Sigma value. Sigma in two target cases (δ represents). The formula for calculation is:

$$\delta = \frac{f_1^2 - f_2^2}{f_1^2 + f_2^2} \quad (9)$$

where f1 and f2 are the feature values for the particle's two targets. In order to eliminate the likelihood that the value of π being similar to 1 or -1 due to the large variation in the target function magnitude, the target function is standardized.

$$f_1^{'} = \frac{f_1 - f_1^{min}}{f_1^{max} - f_1^{min}} \tag{10}$$

$$f_2^{'} = \frac{f_2 - f_2^{min}}{f_2^{max} - f_2^{min}} \tag{11}$$

where $f_1^{min}, f_1^{max}, f_2^{min}, f_2^{max}$ are the maximum and minimum values of the particles on the two objectives , respectively, thus obtaining a new formula for calculating δ:

$$\delta = \frac{f_1^{'2} - f_2^{'2}}{f_1^{'2} + f_2^{'2}} \tag{12}$$

The steps of Sigma method to solve the global extreme value are as follows:

1- Solve the value of each particle in the population and the value of each non-dominated solution set in the outer set ES;

2- Find the particle externally concentrated j closest to the value of the ÿ I Population Particle;

3- Use jth particle solution in the outer set as gbest(i) of the population's ith atom.

The Sigma method is used in the early stage of the algorithm to locate the global extreme point of particle I from the external set, which will cause the algorithm to converge rapidly into the optimum global non-inferior surface. When the algorithm runs to certain algebra, the particles as the global extreme value are on the ideal non-inferior surface. It is appropriate to consider selecting individuals in the dispersed region on the front of Pareto, and to direct the particle swarm to evolve into the dispersed region. Then, the globally optimum individuals are updated with the following methods at the later stage of evolution:

1) If the crowded distance of all individuals in Pareto is infinite, that is, only include a small number of boundary individuals, then randomly select one as *gbest*.

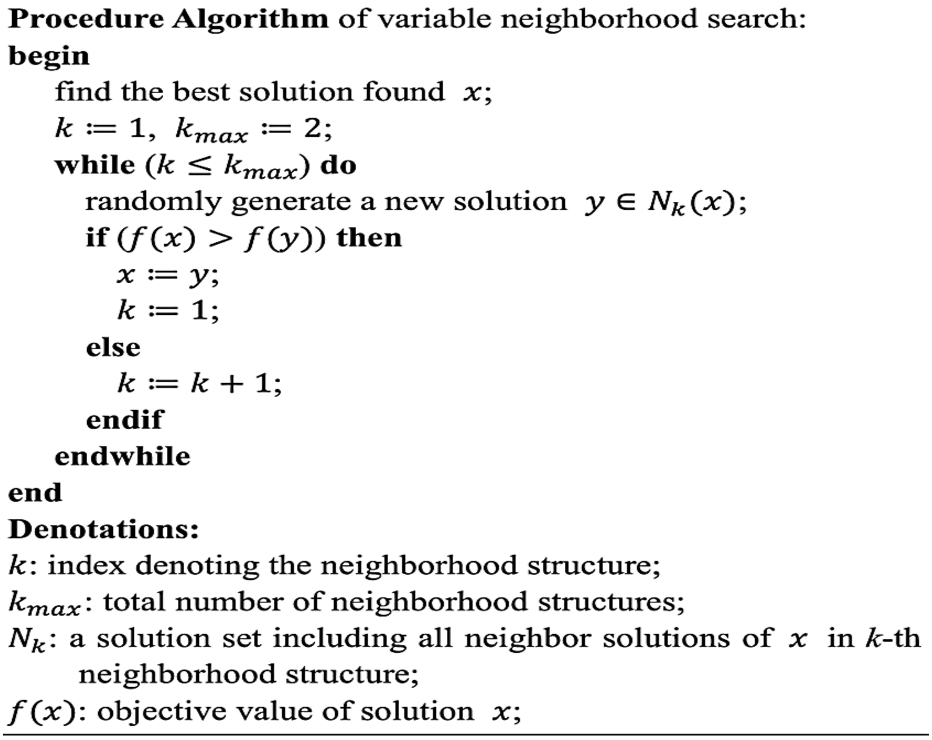
2) If Pareto contains individuals with crowded distances that are not infinite, then use the roulette method to select, that is, choose individuals with larger crowded distances as *Gbest* with a higher probability. The calculation formula is

$$P(i) = \frac{ES[i]_{distance}}{\sum_{i=1}^{popsize} ES[i]_{distance}} \tag{13}$$

where $P(i)$ is the probability that the *ith* individual in the external set is selected, $ES[i]_{distance}$ is the *ith* individual's aggregation distance, and pop size is the number of particles in the current outer range. It should be remembered that the individual aggregation distance includes an infinite assembly that allows the collection of roulette wheels to malfunction, so that the boundary points are not counted by the pop size in the formula.

Variable neighborhood search

Taillard E. [13] suggested a vector search algorithm for the neighborhood, which achieved successful results in many problems. Murata T. [15] integrated genetic algorithms with variable neighborhood search algorithms, and Nawaz et al. [24] tested the distribution of algorithm blends with the search algorithm for variable neighborhoods. The application of the variable neighborhood search algorithm greatly increases the centralized search capabilities of these algorithms, so that the variable neighborhood search algorithm is used for efficient Pareto solution searching for neighborhood. Execution procedure of search algorithm for vector neighborhoods is shown in Figure- 3.



```
Procedure Algorithm of variable neighborhood search:
begin
   find the best solution found $x$;
   $k := 1,\ k_{max} := 2$;
   while ($k \le k_{max}$) do
     randomly generate a new solution $y \in N_k(x)$;
     if ($f(x) > f(y)$) then
       $x := y$;
       $k := 1$;
     else
       $k := k + 1$;
     endif
   endwhile
end
Denotations:
$k$: index denoting the neighborhood structure;
$k_{max}$: total number of neighborhood structures;
$N_k$: a solution set including all neighbor solutions of $x$ in $k$-th
     neighborhood structure;
$f(x)$: objective value of solution $x$;
```

Figure 3-Variable neighborhood search algorithm flow

Algorithm flow

A multi-objective particle swarm optimization algorithm is used in this paper to conduct a decentralized PFSP search. The composition and maintenance strategy of its Pareto solution set is proposed, and a two-stage global optimal selection mechanism is also proposed to effectively lead particles to leading-edge non-dominated solution convergence. To search for further Pareto solutions, a variable neighborhood search algorithm is proposed to perform a clustered neighborhood search on the non-dominated solution obtained by the multi-objective particle swarm algorithm. The hybrid algorithm basic flow algorithm is as follows.

1) Parameters of initialization algorithm: evolutionary population size (pop size), external collection size (ES) popsize, inertia coefficient (w), cognitive coefficient (c1) and social coefficient (c2), high coefficient of dominance (C).

2) Particulate populations are initialized as follows:

(A) Use NEH to produce a 10 percent workpiece processing sequence, and determine the scheduling goal and transform it to a formula-based particle location vector (5);

(B) Randomly produce the location vectors of the remaining 90 percent of the particles, obtain a pair of workpiece processing sequences in conjunction with the ROV law, and determine the two scheduling goals for each particle according to the addition phase sequence;

(A) Randomly initialize vector speeds of all particles in the population;

b) Let the current location be the local equilibrium of each particle, perform a simple sorting on the initial population, and apply the non-dominated solution to the external set ES;

C) Update all extreme values.

1) Loop steps 4)-6) until you have met the stop condition.

2) The following operations apply to all particles:

A) Use equations (6) and (7) to update all particle velocity and position;

b) Determine the workpiece processing sequence corresponding to each particle position vector according to the ROV rule, and calculate two scheduling targets for each particle;

A) Update the external ES set and update each particle's individual extremum.

1) To update the external set ES, perform a variable neighborhood search algorithm for the Pareto solution in the external set ES.

2) Change all extreme values.

3) External output Package ES. The hybrid particle swarm algorithm flow unique algorithm

Simulation And Results Analysis

The data in this paper were based on the 120 benchmark test problems presented by Taillard [14] in 1993, in order to check the new hybrid particle swarm optimization algorithm. The multi-objective mixed ion optimization algorithm proposed in this paper is used to solve the problem, and, under the same hardware conditions, the popular hardware is used. To solve this, the strong dominant evolutionary algorithm (SPEA2) is used, and the results obtained from the two algorithms are compared. The 2015b matlab programming algorithm is used, the computer Processor is Intel core i7, the key frequency is 2.7 G, the physical memory is 8 GB, and the operating system is Windows eight. First, a brief introduction to the strongly dominated evolution algorithm is presented.

Strongly Dominating Evolutionary Algorithm

Zitzler and Thiele [9] suggested a fast evolutionary pareto algorithm (SPEA), based on the 1999 Pareto approach. In 2001, SPEA was strengthened to fix its limitations, and SPEA2 was proposed [7]. Within the SPEA2 algorithm, an intensity value is assigned to each particle in the population and elite group. The strength value contains information about the dominance relation and information about the distance. The ranking value of each particle is determined by the intensity of the individual which controls the current particle. At the same time, the algorithm uses niche strategy to evaluate each particle's density information. The final value of fitness is the sum of the sorted quantity and the particle density. Furthermore, the bidding technique is followed to ensure elite selection size.

Evaluation Index

The number of non-dominated solutions (NPS) measures the number of non-dominated solutions that can be sought by multiple algorithms, which can be compared to the whole Pareto frontier.

1) **The Spacing Metric (SM)** evaluates the distribution of the solution set in the target space by calculating the change in distance between each individual in the solution set and the neighboring individual, and determines its evaluation function as follows:

$$S = \sqrt{\frac{\sum_{i=1}^{N}(\bar{d} - d_i)^2}{N-1}} \qquad (13)$$

where $d_i = \min\left(\sum_{m=1}^{k}\left|f_m^i - f_m^j\right|\right), i, j = 1,2, \dots, N, i \neq j$; N is the number of individuals in the solution set; $\bar{d}$ is the average value of alld_i; the smaller the value of S, the more uniform the distribution of the solution set.

2) **Diversification metric (DM).** This metric helps to measure the solution set's extensibility. The absolute Euclidean distance between the person and other entities is calculated by this metric. The evaluation role is described as follows:

$$d_i = \sqrt{max\sum_{i=1}^{N}(\|x_i - y_i\|)} \qquad (15)$$

where d_i is the Euclidean distance between non-dominated solutions x_i and y_i.

3) **Relative percentage increase indicator (RPI).** This indicator is used to judge the relative growth rate of the solution from the ideal point in the solution. The ideal point is the solution consisting of a single target optimal value in the algorithm, and the RPI is the growth rate of each solution in the solution set from the ideal point. The Ideal Solution average value sets the makepan value to min(MS) and to min(TFT) for the optimal solution. The I Pareto makespan value is expressed as MS(S(i)) and the TFT value is expressed as TFT(S(i)); the relative growth ratio indicator function is then defined as:

$$RPI = \frac{\sum\left[w_1 * \left(MS(S(i)) - \min(MS)\right) + w_2 * \left(TFT(S(i)) - \min(TFT)\right)\right]}{N} \qquad (16)$$

where N is the number of solutions in the Pareto solution set.

Experimental Parameter Settings

The experimental parameters of this equation are calculated as follows: c1 = c2=2.0, the initial value of the inertia coefficient w is calculated to 0.9, β = 0.975, the minimum cannot be less than 0.4, the particle's minimum position value xmin = 0, the particle's maximum position value xmax = 4.0, the particle’s minimum velocity value vmin = −4.0, the particle's maximum velocity value vmax = 4.0. Every instance runs 10 times autonomously and continuously.

The SPEA algorithm's experimental parameters are set as follows: the initialization method is the same as the algorithm in this article, using the NEH and random initialization combination, using the competition selection strategy, the crossover operator uses OX crossover, the mutation method is the interchange mutation, the crossover probability is set to 0.8, and the mutation probability is set to 0.8. The population size is 40 and the maximum number of iterations is 500.

Comparison and Analysis of Experimental Results

This paper picks 30 questions from the 20 workpiece series in the TA test set issue for review and compares the sample data from the http:/mistic.heig-vd.ch/taillard/ Taillard website. The combined multi-objective particle swarm in this article MPSO-VNS describes the algorithm, and the two indicators are contrasted with the four NPS, SM, DM, and RPI indicators. Experimental results using technical benchmarks problems were conducted and proved the effectiveness of the hybridization and the advantage of HPSO algorithm compared to other local search algorithms in the resolution of the scheduling problem. The results of the experimental comparison are set out in Table2.

According to the experimental comparison results, the multi-objective mixed particle swarm optimization algorithm proposed in this paper is better than the SPEA2 algorithm in most instances. The results are better in various indicators, and more non-dominated solutions could be obtained. The distribution, extensibility, and relative growth ratio indicators have also been improved to a certain extent, thus confirming the effectiveness of the algorithm.

Figure 4 shows the comparison of the Pareto solution set provided by the two algorithms to solve the ta021 problem. It can be shown that the Pareto solution set obtained by MPSO-VNS is relatively well distributed and convergent.

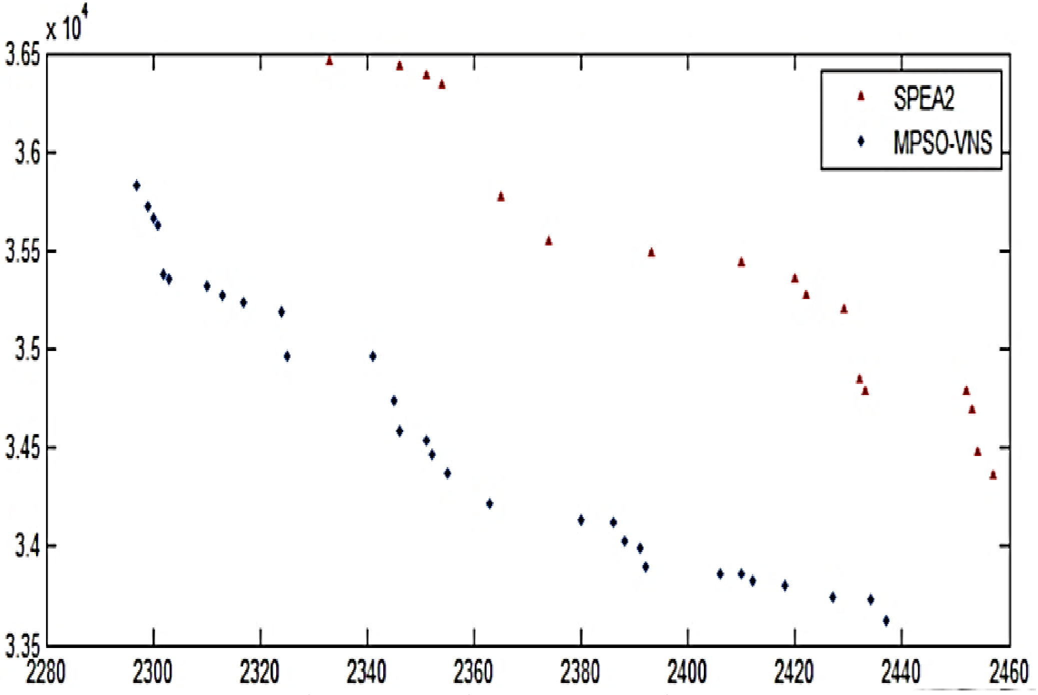


Figure 4-Comparison of the results of two algorithms for solving the ta021 problem

Table 2-Experimental comparison results

Benchmark	Problems description		NPS		SM		DM		RPI	
	N	M	SPEA 2	MPSO -VNS	SPEA 2	MPSO -VNS	SPEA 2	MPSO -VNS	SPEA 2	MPSO -VNS
Ta001	20	5	13	34	40.21	16.11	17.29	15.91	0.0255	0.0077
Ta002	20	5	39	17	28.30	19.65	46.43	23.88	0.0168	0.0058
Ta003	20	5	10	56	49.47	27.49	16.02	29.82	0.0279	0.0093
Ta004	20	5	34	58	23.23	15.78	18.98	25.26	0.0209	0.0058
Ta005	20	5	35	37	46.84	29.72	38.17	18.59	0.0175	0.0002
Ta006	20	5	36	37	23.19	26.90	19.11	20.02	0.0245	0.0012
Ta007	20	5	12	30	31.23	22.89	52.37	22.43	0.0264	0.0086
Ta008	20	5	22	56	43.17	24.33	52.20	30.31	0.0297	0.0048
Ta009	20	5	18	31	50.23	18.70	48.49	17.16	0.0000	0.0084
Ta010	20	10	34	20	18.44	22.20	21.34	28.18	0.0260	0.0021
Ta011	20	10	23	50	57.31	30.03	45.95	25.42	0.0184	0.0055
Ta012	20	10	38	32	50.92	23.74	38.92	21.09	0.0297	0.0063
Ta013	20	10	15	26	37.43	23.50	59.05	18.60	0.0158	0.0003
Ta014	20	10	18	33	35.18	18.48	44.74	21.47	0.0144	0.0061
Ta015	20	10	14	19	35.90	22.90	51.27	22.70	0.0240	0.0036
Ta016	20	10	14	21	29.98	24.61	35.42	16.70	0.0068	0.0005
Ta017	20	10	36	58	38.44	25.62	34.55	24.64	0.0149	0.0049
Ta018	20	10	27	58	38.11	21.86	52.94	18.03	0.0270	0.0019
Ta019	20	10	27	41	52.26	20.81	18.42	21.07	0.0172	0.0012
Ta020	20	20	14	17	51.41	30.58	21.98	24.32	0.0254	0.0021
Ta021	20	20	36	25	44.59	15.18	22.30	19.53	0.0222	0.0015
Ta022	20	20	29	31	32.26	29.24	32.70	19.65	0.0176	0.0019
Ta023	20	20	20	52	52.60	29.89	53.67	24.41	0.0074	0.0004
Ta024	20	20	25	15	39.71	27.03	51.54	19.82	0.0200	0.0064
Ta025	20	20	22	16	31.22	16.49	17.70	28.72	0.0025	0.0028
Ta026	20	20	12	22	58.12	19.17	33.67	30.97	0.0188	0.0054
Ta027	20	20	17	44	55.30	20.98	39.18	26.53	0.0198	0.0070
Ta028	20	20	13	48	40.32	25.71	34.13	20.33	0.0219	0.0050
Ta029	20	20	15	44	43.42	17.50	46.00	24.11	0.0267	0.0054
Ta030	20	20	17	35	42.51	26.47	43.17	16.61	0.0295	0.0045

Conclusions

A multi-objective hybrid optimization algorithm, combining particle swarm optimization algorithm and variable neighborhood search algorithm for the multi-objective PFSP problem, was proposed in this paper. NEH heuristic algorithm was used to initialize the population. The results showed remarkable improvement in several aspects, that include: the quality of the initial solution; application of continuous PSO algorithm for effective global search, based on the ROV rule; introduction of non-dominated solutions for external collection management; implementing a policy of high dominance and aggregation distance to efficiently preserve external collections; implementing a two-stage global optimal selection method to fairly direct population convergence; using a vector neighborhood search algorithm to conduct a clustered search on an external range of Pareto solution that improves the central search capability of the algorithm. The algorithm integrates the global search functionality of the PSO algorithm with the local search functionality of the vector neighborhood search algorithm to maintain an efficient balance between localized search and centralized search, which significantly increased the algorithm's search capability. The Taillard benchmark problem is solved using the dual algorithm and the test results are compared with the SPEA2 algorithm. For this paper, the algorithms obtained good results when its effectiveness was tested.

References

1. Pinedo ML. **2008**. *Scheduling: theory, algorithm, and systems*, 3rd edn. Springer, Berlin.
2. Graham RL, Lawler EL, Lenstra JK, Rinnooy Kan AHG **1979**. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Ann Discrete Math*, **5**: 287–326.
3. Alaa, S. and Hanan A. **2019**. Exact methods for Solving a Multi-objective problem on single machine Scheduling, *Iraqi Journal of Science,* **60**(8): 1802-1813.
4. Dugardin F., Yalaoui F. and Amodeo L. **2010**. New multi-objective method to solve re-entrant hybrid flow shop scheduling problem. *Eur J Oper Res*, **203**: 22–31.
5. Karimi N, Zandieh M. and Karamooz HR. **2010**. Bi-objective group scheduling in hybrid flexible flowshop: a multi-phase approach. *Expert Syst Appl*, **37**: 4024–4032
6. Carlos Coello, Gary Lamont, David Veldhuizen. **2007**. *Evolutionary algorithms for solving multi-objective problems,* 2nd edn, vol 2. Springer, New York, pp 5–60.
7. Hanan A.chachan **2020**. Strength Pareto evolutionary algorithm II for Multi-objective Knapsack problem. *international journal of psychosocial Rehabilitation* , **24**(01),ISSN:1475-7192
8. Abbas, W. A. **2020**. Genetic Algorithm-Based Anisotropic Diffusion Filter and Clustering Algorithms for Thyroid Tumor Detection. *Iraqi Journal of Science*, 1016-1026.
9. Zitler E, Thiele L. SPEA2. **2001**. Improving the strength pareto evolutionary algorithm for multiobjective optimization. Eurogen -Evolutionary methods for design, optimization and control with applications to industrial problems.
10. Mostaghim S, Teich J. **2003**. Strategies for finding good local guides in multi-objective particle swarm optimization. Proc of the 2003 IEEE Swarm Intelligence Symposium. Indianapolis: IEEE Service Center :26−33.
11. Wang W. and Farid H. **2009**. Exposing digital forgeries in video by detecting double quantization. Proc. of the 11th ACM workshop on Multimedia and security. New York,USA : 39−48.
12. Wang W. and Farid H. **2006**. Exposing digital forgeries in video by detecting double MPEG compression. Proc. of the 8th Workshop on Multimedia and Security.New York. 37−47.
13. Taillard E. **1993**. Benchmarks for basic scheduling problems. *European Journal of Operational Research*, **64**: 278− 285.
14. Jawad, A. A., Ali, F. H., & Hasanain, W. S. **2020**. Using Heuristic and Branch and Bound Methods to Solve a Multi-Criteria Machine Scheduling Problem. *Iraqi Journal of Science*, 2055-2069.
15. Carlos Coello, Gary Lamont, David Veldhuizen **2007**. *Evolutionary algorithms for solving multi-objective problems*, 2nd edn, vol 2. Springer, New York, pp 5–60.
16. Murata T., Ishibuchi H. and Tanaka H. **1996**. Multi-objective genetic algorithm and its applications to flow shop scheduling. *Comput Ind Eng*, **30**: 957–968.

17. Murata T., Ishibuchi H., Tanaka H. **1996**. Multi-objective genetic algorithm and its applications to flow shop scheduling. *Comput Ind Eng*, **30**: 957–968.

18. Makinde, T. A. and Daniel, D. O. **2020**. Modal Analysis of Vibration of Euler-Bernoulli Beam Subjected to Concentrated Moving Load. *Iraqi Journal of Science*, 2324-2334.

19. Chang PC, Hsieh JC. And Lin SG. **2002**. The development of gradual-priority weighting approach for the multiobjective flowshop scheduling problem. *Int J Prod Econ*, **79**: 171–183

20. Neppalli VR, Chen CL. And Gupta JND. **1996**. Genetic algorithms for the two-stage bi-criteria flow shop problem. *Eur J Oper Res*, **95**: 356–373.

21. Chang PC, Hsieh JC. and Lin SG. **2002**. The development of gradual-priority weighting approach for the multiobjective flowshop scheduling problem. *Int J Prod Econ*, **79**: 171–183.

22. Deb, K. and Agrawal, R.B. **1995**. “Simulated binary crossover for continuous search space,” *Complex Syst.*, **9**(2): 115–148.

23. Deb, K. and Pratap, A. **2002**. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Trans. Evol. Comput.*, **6**(2): 182–197, Apr.

24. Nawaz M, Enscore E. and Ham I. **1983**. A heuristic algorithm for the m-machine n-job flow shop sequencing problem. *Omega*, 1983, **11**(1): 11–95.