2nd INTERNATIONAL CONFERENCE ON
RESEARCH IN SCIENCE AND TECHNOLOGY
Istanbul-Turkey 14 March 2016
RSTCONF
دومین کنفرانس بین المللی
پژوهش در علوم و تکنولوژی
ترکیه - استانبول ۲۴ اسفند ۱۳۹۴

Electromagnetism-like algorithm for permutation flowshop

Kamran Zamanifar,
Department of computer engineering Isfahan University Isfahan, Iran.
zamanifar@eng.ui.ac.ir

Davood Fallahi
Department of computer engineering Payame Noor University, Tehran, Iran.
d_fallahi@pnu.ac.ir

Abstract:
Problem of scheduling in permutation flowshop is dealt with by minimizing the makespan. If there are more than three machines, then it is NP-compelet problems and should be solved by metaheuristic algorithm. The objective of this article was to minimize total flowtime of jobs. To this end, the algorithm of electromagnetism-like method (EM) was applied, which utilized an attraction-repulsion mechanism to move the sample points toward optimality. The computational results demonstrated that EM was robust in practice. Moreover, this algorithm had two other advantages: i: simplicity of application and ii: occupying minor memory capacity. In the first stage, the base algorithm was developed by compensating for the lack of convergence. In the second stage, the parallel algorithm was applied. Finally, results of the algorithm were compared with those of other algorithms.
Keywords: Permutation flowshop; Scheduling; Makespan; Total flowtime; Electromagnetism

1. Introduction

In a flowshop scheduling problem (FSP), there are n independent jobs {J1 , . . . , Jn } which should be processed on m different machines {M1 , . . . , Mm }. Each job is composed of m operations and every operation needs a different machine. Oij refers to the operation on machine i of job j. A processing time Pij is required for every operation. In a flowshop, all jobs have the same processing order on machines.

Because of following a technological order on the machines for different jobs, the objective is to find an ordering of the jobs on the machines or a sequence which could optimize a considered criterion. Thus far, the most common criterion has been minimization of total completion time of the schedule, which is often referred to as makespan (Cmax). Also, some other interesting variations exist for this problem (see [1] or [2]):

• All operations are independent and available for processing at time 0.
• All m machines are continuously available.
• Each machine i can process at most one job j at a time.
• Each job j can be processed only on one machine i at a time.
• Processing of a given operation Oij cannot be interrupted; i.e. no preemption is allowed.
• Setup and removal times are sequence-independent and included in the processing times; otherwise, they are negligible and can be ignored.
• In-process inventory is allowed. If a given operation needs an unavailable machine, then the operation joins an unlimited queue on that machine.

Following the four parameter notation A/B/C/D by Conway et al. [3], the problem can be classified as n/m/F
Fmax.
Graham et al. [4] proposed a more recent three-parameter notation (/ /)and the problem was denoted as F //Cmax.

FSP is known to be NP-complete in the strong sense when m >3 (see [5]); if m=2, Johnson's algorithm [6] achieves an optimal solution in polynomial time. In general, (n!)m schedules have to be assumed ((n!)M−2 for Cmax criterion).

This paper simplified FSP, which
was permutation FSP or PFSP. In the PFSP, job passing is not permitted; in other words, processing sequence of the jobs is the same for all the machines. Accordingly, n! schedules are possible and the problem is then denoted as n/m/P /Fmax or as F /prmu/Cmax (see [2]).

2. Available methods for the PFSP

Generally speaking, the PFSP can be solved using either exact or heuristic methods. The former methods are only practicable in small instances (less than 15–20 jobs); even in that case, solution times tend to be high. However, some types of exact techniques can be applied for obtaining optimal solutions for large instances when starting from the high quality, near-optimal solutions obtained by advanced metaheuristics. Heuristic approaches have received great focus of research efforts. Heuristics for the PFSP can be divided into constructive and improvement methods; the former are techniques which construct a feasible schedule from scratch and the latter are algorithms that seek to improve a previously generated schedule. Many constructive heuristics are available (for a comprehensive review, refer to [4,5,6,7,8,9,10,11]). NEH is a contractive heuristic as the best type of the algorithm concerning time of solution [4]. Its improvement methods can be found in [12,13]. Metaheuristics method is an improved heuristic method, in which tabu search, genetic algorithm, ant-colony and simulated annealing and other hybrid method are observed. First,

simulated annealing method was proposed in [14,15] and then ant-colony was introduced in [20]. Genetic algorithm was the next one which was proposed in [21,22,23,24,25]. The focus of this paper was on the electromagnetism-like method (EM) [27] as an optimality metahuristic method, which is used for solving the problems that cannot be solved within a reasonable time.

2. Formulating the permutation flowshop

Scheduling the problem

n is total number of jobs which should be scheduled, m is total number of machines in the flowshop, ∂ is the ordered set of jobs which are already scheduled out of n jobs (partial sequence), q(∂,j) is completion time of partial sequence ∂ on machine j (i.e. release time of machine j after processing all the jobs in partial sequence ∂),and q(∂i,j) is completion time of job i on machine j when the job is appended to partial sequence ∂.

To calculate the start and completion times of jobs on machines in permutation flowshops, recursive equations are applied:

Initialize q(∂i,0), as completion time of job i on machine 0, equal to zero. This time is availability time of a job in the flowshop and is equal to 0 for all the jobs in case of static flowshops.

For j=0 to m do

q(∂i,j)=max(q(∂,j); q(∂i,j-1))+Pij

For time of job i, Ci is given by:
Ci = q(∂i,m)

After scheduling all jobs, total flowtime F and makespan M are obtained as follows:

$$\mathbf{F=} \sum_{i=1}^{n} Cji$$

and

M = max {Ci , i = 1 , 2, ….., n }

It is worth noting that q(ϕ,j) is equal to 0 for all j, where ϕ is a null schedule.

To calculate start and completion times of jobs on machines in permutation flowshops, recursive equations are applied:

Initialize qðri; 0Þ, as completion time of job i on machine 0, equal to zero, which indicates availability time of a job in the flowshop and is equal to 0 for all the jobs in case of static flowshops.

3. Explaining electromagnetic algorithm for solution (pfsp)

Sample point toward optimality: In this algorithm, any response is considered a charged particle. These particles that are more optimized have more change and attract other particles. On the other hand, those particles that are less optimized repulse other particles.

Main idea of this article was that better points might exist around the optimal point. thus, weak points were moved toward the optimal point.

ALGORITHM 1. EM(m)

1: Initialize()
2: while termination criteria are not satisfied do
3: Local()
4: CalcF()
5: Move()
6: end while

Em algorithm is used for solving the problems which require optimality and applies attraction-repulsion mechanism to move ….
In algorithm 1, first, an initial population is produced, which can be done in two methods: either by random method or by NEH method. The second method which applies the NEH method is much better than the first one. The FRB [29] method which was used in this article was more optimized than the two previous methods.
(see the following code).

ALGORITHM 2. FRB
Procedure FRB

Calculate $Pj = \sum_{i=1}^{m} pij, \forall n \in N$

Sort Pj in a decreasing order:
π := ∅
For step1:= 1 to n
j := job(Pj [step])
Test job j in all possible positions of π% (Taillard's accelerations) Insert job j in π in position p resulting in the for the lowest Cmax.
Step 2: = max(1, p − k) to min(step, p + k)
Extract job h in position of step2 from π
Test job h in all possible positions of π %
(Taillard's accelerations)
Insert job h in π in the position resulting in the lowest Cmax.
Endfor
Endfor
End.

P: poison of the new job
K: poison around the new job from outer loop
and π: partial sequence
FRB is an NEH algorithm, which is the same as NEH with only one difference that only one inner loop has been added to it. All the jobs between (p-k…p+k) are tested in this inner loop to achieve the best sequence. The idea behind this method is that these jobs which are farther than the newly entered point in the outer loop denoted greatly affect the optimal answer. However, by changing the point around the new point, a better answer may be finally obtained. In this algorithm, if k value is assumed lower, the algorithm complexity would be equal to the NEH complexity.
In the next stage, a series of tasks is performed for discovering the answers. Therefore, a local search is applied over all or some points of the population, assuming that a better optimized point exists around them.

This local search is done at some specific points.
It is possible to perform this search over all the points; but, speed of the process is reduced.
In this article, a local search which searched around a specific point (as explained it in the **cacf()** function and called perturbation) was used.
The perturbation point goes out of the local region and enters another feasible region. Afterward, a local search is started around this point and the existing point is replaced with those points with better answers and so on.

Afterward, the exerted force on each of the elements of the population is computed. After computing the force, each of the elements is moved in the direction of the exerted force.
In this algorithm, every answer is considered a point in n-dimensional space. Value of each dimension should be based on the following condition.
including: Ln< Xn <Un ;
n dimension of the problem,
Uk upper bound in the kth dimension,
Lk lower bound in the kth dimension, and
f (x) points to the function to be minimized.
To compute value of the exerted force, the particle charge must be computed as follows:

$$q^{i} = \exp\left(n \frac{f(x^{i}) - f(x^{best})}{\sum_{n=1}^{m} f(x^{h}) - f(x^{best})}\right)$$

In each iteration, charges of the points were computed according to values of their objective function. However, in the present heuristic, charge of each point was not constant and changed from one iteration to another.

Charge of each point i, q i, determines point I's power of attraction or repulsion.

Accordingly, the points with better objective values possessed higher charges. The fraction was multipled by dimension n because, in higher dimensions, the number.
of points in the population tended to get large. Consequently, the fraction might become very small and cause overflow problems in calculating the exponential function.

Note that, unlike electrical charges, no signs were attached to the charge of an individual point in Eq. (2). Instead, direction of a particular force between two points was determined after comparing their objective function values. Hence, total force F i exerted on point i is computed by Eq. (2).

$$F^{i}_{j} = \left\{ \begin{array}{l} (x^{j} - x^{i}) \dfrac{q^{i} q^{j}}{\left\| x^{j} - x^{i} \right\|^{2}} \\ (x^{i} - x^{j}) \dfrac{q^{i} q^{j}}{\left\| x^{j} - x^{i} \right\|^{2}} \end{array} \right\} \qquad (2)$$

As seen in Eq. (2), among the two points, the point that had a better objective function value attracted the other one. Contrarily, the point with the worse objective function value repelled the other.

In calculating the force, if the algorithm had premature convergence, none of the feasible points were searched for and the optimal point might not be found. In order to prevent premature convergence, Eq. (2) was modified:

In order to preclude premature convergence, the current population was somehow "perturbed" so that at least one of the points had a chance to move to the possibly omitted parts of the feasible region. Hence, one of the points in the population other than the currently best point was selected as the "perturbed point", as denoted by xp (refer to Eq. (3)).

$$x^p = \arg\max\left\{\left\|x^{best} - x^i\right\|, i = 1,2,...m\right\}$$

The perturbed point (xp) was selected as the farthest point from the currently best point while λ is uniformly distributed between 0 and 1 and is used for perturbating the component force

$$F_j^p \begin{cases} (x^j - x^p)\dfrac{\lambda q^p q^j}{\left\|x^j - x^p\right\|^2}, iff\ (x^j) \langle f(x^p) \\ (x^p - x^j)\dfrac{\lambda q^p q^j}{\left\|x^j - x^p\right\|^2}, iff\ (x^p) \le f(x^j) \end{cases}$$

Finally, total force vector Fi exerted on each point was calculated by adding the individual component forces; i.e.:

$$F^i = \sum_{j \neq i}^{m} F_j^i, i = 1,2,..., m.$$

Refer to algorithm 2:
if the random variable λ was less than the parameter *v*, direction of the component force would be reversed. Consequently, one point existed in the population, for which direction of movement might be reversed.

ALGORITHM 2 CalcF ()

1. $x^p \leftarrow \arg\max\left\{\left\|x^{best} - x^i\right\|, i = 1,2,..., m\right\}$

2. *for* $i = 1$ *to* m *do*

3. $q^i \leftarrow \exp(-n\dfrac{f(x^i) - f(x^{best})}{\sum_{n-1}^{m}(f(x^i) - f(x^{best}))}$

4. $F^i \leftarrow 0$

5. *endfor*

6. *for* $i = 1$ *to* m *do*

7. *for* $j = 1$ *to* m *do*

8. *if* $i \neq j$ *then*

9. *if* $x^i \neq x^p$ *then*

10. $F_j^i \leftarrow (x^j - x^i)\dfrac{q^i q^j}{\left\|x^j - x^i\right\|^2}$

11. *else*

12. $\lambda \leftarrow U(0,1)$

13. $F_j^i \leftarrow (x^j - x^i)\dfrac{q^i q^j}{\left\|x^j - x^i\right\|^2}$

14. *if* $\lambda \langle v$ *then*

15. $F_j^i \leftarrow -F_j^i \{\text{Re}\, verseDirection\}$

16. *endif*

17. *endif*

18. *if* $(f(x^j) \langle f(x^i)$ *then*

19. $F^i \leftarrow F^i + F_j^i \{Attraction\}$

20. *else*

21. $F^i \leftarrow F^i - F_j^i \{\text{Re}\, pulsion\}$

22. *endif*

23. *endif*

24. *endfor*

25. *endfor*

4. Moving along total force vector (move)

After evaluating total force vector F i, point i was moved in the force direction by a random step length as in Eq. (6). In this equation, the random step length is assumed uniformly distributed between 0 and 1. RNG denotes the allowed range of movement toward lower bound lk or upper bound uk for the corresponding dimension.

$$x^i = x^i + \lambda \frac{F^i}{\left\|F^i\right\|}(RNG) \qquad i = 1,2,...,m$$

Algorithm 3 demonstrates the move procedure. Note that, the best point, Xbest , is not moved and is carried to the subsequent iterations.

ALGORITHM 3 Move

1. *for* $i = 1$ *to* *do*

2. *if* $i \neq best$ *then*

3. $\lambda \leftarrow U(0,1)$

4. $F^i \leftarrow \dfrac{F^i}{\left\|F^i\right\|}$

5. *for* $K = 1$ *to* *do*

6. *if* $F_k^i \rangle 0$ *then*

7. $x_k^i \leftarrow x_k^j + \lambda F_k^i (u_k - x_k^i)$

8. *else*

9. $x_k^i \leftarrow x_k^j + \lambda F_k^i (x_k^i - l_k)$

10. *endif*

11.*endif*

12.*endfor*

13.*endfor*

Electromagnetism as a parallel algorithm one of the factors of hawing improved result is that larger population should exist. By increasing size of the initial population, achieving optimal result would be changed.
EM the end: population growth would have a negative effect on speed of the process.
An alternative solution for speed improvement of the process and optimization of the algorithm efficiency is through applying parallel populations.
In this method, a large population is divided into minor ones, which has three advantages:

- A higher volume of feasible region is searched.
- Each of the minor populations can be processed by a separate processor, giving the advantage of applying parallel processing.
- The algorithm speed is increased.

If the population has p elements, it is divided to groups of M/7. By applying the algorithm to each of the groups, the answer would move toward an optimal condition.
in groups of M/7 . by applying algorithm an each of the groups , an answer more teward.

5. Computational experience

In this section, the proposed EM is compared with other simulated annealing, tabu search and some other state-of-the-art techniques for the PFSP.

The compared methods included the NEH heuristic by Nawaz et al. enhanced by Taillard [22] (NEHT), GA by Chen et al. (GAChen), simulated annealing by Osman and Potts (SAOP), tabu search by Widmer and Hertz (Spirit), GA by Reeves (GAReev), hybrid GA by Murata et al. (GAMIT), iterated local search procedure by Stützle (ILS), GA by Aldowaisan and Allahvedi adapted to PFSP (GA_AA) and finally the recent ant-colony algorithms by Rajendran Ziegler [27] referred to as M-MMAS.

For the evaluation, the well-known standard benchmark set by Taillard [8] which was composed of 120 different problem instances ranging from 20 jobs and 5 machines to 500 jobs and 20 machines was used. The benchmark contained 10 repetitions for each of the considered combinations of n and m. The results were averaged for all the 10 instances in a given combination.

For evaluating different methods, a similar performance measure was applied by the following equation:

$$\sum_{i=1}^{R}\left(\frac{Heu_{soli}-Best_{sol}}{Bestsol}\times 100\right)/R$$

Where Heusoli is the solution given by any of the R repeti-
tions of the considered algorithms; in this case, Best sol is either the optimum solution or the lowest known upper bound for Taillard's instances as of late April 2007 (these values are available in Taillard [45]).
In Table 1, Em algorithm shows the ability to compete with the mentioned algorithms. Moreover, better results were achieved, especially in the case of obtaining a better answer.

6-Conclusions

In this paper, first, the related literature of PFSP was reviewed and then its formulation was stated. Afterward, the FRB algorithm was used for the initial populations.

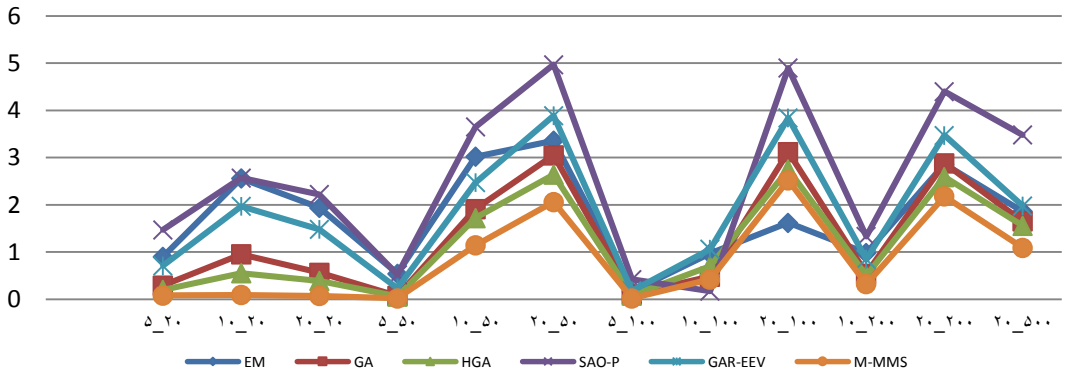
The electromagnetic algorithm was presented and the base algorithm was improved. Paralleling the algorithm and solving problems through this algorithm were performed afterward. Finally, it was changed on Taillared's benchmark.

This algorithm moved all the points monotonously toward the optimal point. In the initial population section, it was distributed in the whole feasible region. Then, a local search was used, which had to be a monotonous one due to searching all the areas around all the points. Accordingly, the effect of the answer was observed. Monotonous initial population and local search on this issue will be investigated in future.

Table۱

Average relative percentage deviation (RPD) over optimum solution or lowest known upper bound for Taillard's instances obtained by the evaluated methods

n-m	**EM**	GA	HGA	SAO-P	GAR-EEV	M-MMS	ILS	NEH-T	SPIRIT	GAC-HEN	GAM-IT	GA_AA
20_5	0.90	0.29	0.2	1.47	0.71	0.08	0.29	3.35	5.22	3.67	3.28	0.08
20_10	2.56	0.95	0.55	2.57	1.97	0.09	1.26	5.2	5.86	5.03	5.53	1.41
20_20	1.94	0.56	0.39	2.22	1.48	0.07	1.04	3.73	4.58	4.33	4.33	1.37
50_5	0.54	0.07	0.06	0.52	0.23	0.02	0.12	0.84	2.03	1.96	1.96	0.37
50_10	3.02	1.91	1.72	3.65	2.47	1.14	2.38	5.12	5.88	6.25	6.25	3.35
50_20	3.36	3.05	2.64	4.97	3.89	2.06	4.19	6.2	7.21	7.53	7.53	4.52
100_5	0.16	0.1	0.08	0.42	0.18	0.02	0.12	0.46	1.06	1.33	1.33	0.24
100_10	0.97	0.48	0.7	0.17	1.06	0.42	0.85	2.13	5.07	3.66	3.66	1.61
100_20	1.62	3.12	2.75	4.9	3.84	2.52	3.92	5.11	10.2	9.7	9.7	4.73
200_10	0.98	0.54	0.5	1.33	0.85	0.32	0.54	1.43	9.03	6.47	6.47	1.1
200_20	2.86	2.88	2.59	4.4	3.47	2.18	3.34	4.37	16.2	14.56	14.56	4.2
500_20	1.86	1.65	1.56	3.48	1.98	1.09	1.82	2.24	13.6	12.47	12.47	1.98



Reference

[1] Baker KR. Introduction to sequencing and scheduling. New York: Wiley; 1974.
[2] Pinedo M. Scheduling: theory, algorithms and systems. 2nd ., Englewood Cliffs, NJ: Prentice Hall; 2002.
[3] Garey MR, Johnson DS, Sethi R. The complexity of flowshop and jobshop scheduling. Mathematics of Operations Research 1976;1(2):117–29.
[4] Johnson SM. Optimal two- and three-stage production schedules with setup times included. Naval Research Logistics Operational Research 2005;165:479–94. Quarterly 1954;1:61–8.
[5] Ruiz R, Maroto C. A comprehensive review
[6] Palmer DS. Sequencing jobs through a multi-stage process in the minimum total time—a quick method of obtaining a near optimum. Operational Research Quarterly 1965;16(1):101–7.
[7] Gupta JND. A functional heuristic algorithm for the flowshop scheduling problem. Operational Research Quarterly 1971;22(1):39–47.
[8] Campbell HG, Dudek RA, Smith ML. A heuristic algorithm for the *n* job, *m* machine sequencing problem. Management Science 1970;16(10):B630–7.
[9] Dannenbring DG. An evaluation of flow shop sequencing heuristics. Management Science 1977; 23(11):1174–82.
[10] Nawaz M, Enscore EEJ, Ham I. A heuristic algorithm for the *m*-machine, *n*-job flow-shop sequencing problem. OMEGA, the International Journal of Management Science 1983;11(1):91–5. Evaluation of permutation flowshop heuristics, European Journal
[11] Koulamas C. A new constructive heuristic for the flowshop scheduling problem. European Journal of Operational Research 1998;105:66–71.
[12] Davoud Pour H. A new heuristic for the *n*-job, *m*machine 2001;12(7):648–53.23] Reeves CR. A
[13] flow-shop problem. European Journal of Operational Research1991;52:194–202.
[14] Osman IH, Potts CN. Simulated annealing for permutation flow-shop scheduling. OMEGA, The International Journal of Management Science 1989;17(6):551–7.
[15] Ogbu FA, Smith DK. The application of the simulated annealing algorithms to the solution of the *n/m/C*max flowshop problem. Computers & Operations Research 1990;17(3):243–53.
[16] Widmer M, Hertz A. A new heuristic method for the flow shop sequencing problem. European Journal of Operational Research 1989;41:186–93.
[17] Taillard E. Some efficient heuristic methods for the flow Research Society 1993;44(4):375–82. shop sequencing problem. European Journal of Operational Research 1990;47:67–74.
[18] Reeves CR. Improving the efficiency of tabu search for machine scheduling problems. Journal of the Operational
[19] Nowicki E, Smutnicki C. A fast tabu search algorithm for the permutation flow-shop problem. European Journal of Operational Research 1996;91:160–75.
[20] Rajendran C, Ziegler H. Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs. European Journal of Operational Research 2004;155:426–38.

[21] Chen C-L, Vempati VS, Aljaber N. An application of genetic algorithms for flow shop problems. European Journal of Operational Research 1995;80:389–96.
[22] Goldberg D, Lingle Jr R. Alleles, loci, and the traveling salesman problem. In: Grefenstette JJ (Ed.), Proceedings of the first international conference on genetic algorithms and their applications, Hillsdale, NJ, 1985. Lawrence Erlbaum associates, pp. 154–9. 2001;12(7):648–53.
[23] Reeves CR. A genetic algorithm for flowshop sequencing. Computers & Operations Research 1995;22(1):5–13.
[24] Ponnambalam SG, Aravindan P, Chandrasekaran S.Constructive and improvement flow shop scheduling heuristics: an extensive evaluation. Production Planning and Control 2001;12(4):335–44.
[25] Aldowaisan T, Allahvedi A. New heuristics for no-wait flowshops to minimize makespan. Computers & Operations Research 2003;30:1219–31.
[26] Reeves C, Yamada T. Genetic algorithms, path relinking, and the flowshop sequencing problem. Evolutionary Computation 1998;6(1):45–60.
[27] Birbil, S. I. and S. C. Fang, "Electromagnetism-like mechanism for global optimization," *Journal of Global Optimization*, Vol. 25, pp. 263-282 (2003).
[28] Nawaz M, Enscore EEJ, Ham I. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem.
[29] Shahriar Farahmand Rad1, Rubén Ruiz2∗, Naser Boroojerdian1 New High Performing Heuristics for Minimizing Makespan in Permutation Flowshops.
[30] S. ˙ILKER B˙IRB˙IL1, SHU-CHERNG FANG, RUEY-LIN SHEU On the Convergence of a Population-BasedGlobal Optimization Algorithm 1983.