Int J Adv Manuf Technol (2006) 30: 112–117
DOI 10.1007/s00170-005-0029-6

ORIGINAL ARTICLE

Deming Lei · Zhiming Wu

Crowding-measure-based multiobjective evolutionary algorithm for job shop scheduling

Received: 7 July 2004 / Accepted: 12 March 2005 / Published online: 8 October 2005

Abstract Multiobjective evolutionary algorithm (MOEA) has attracted much attention in the past decade; however, the application of MOEA to practical problems such as job shop scheduling is seldom considered. In this paper, crowding-measure-based multiobjective evolutionary algorithm (CMOEA) is first designed, which makes use of the crowding measure to adjust the external population and assign different fitness for individuals; then CMOEA is applied to job shop scheduling to minimize makespan and the total tardiness of jobs. Finally, the comparison between CMOEA and SPEA demonstrates that CMOEA performs well in job shop scheduling.

Keywords Crowding measure · Job shop scheduling · Multiobjective evolutionary algorithm

1 Introduction

Multiobjective evolutionary algorithm (MOEA) has gone through a stagnation period from 1985 to 1994 and the developing stage from 1994 to now since the vector evaluated genetic algorithm (VEGA) was first designed by Schaffer [1] in 1985. Much attention has been paid to apply evolutionary algorithm (EA) to the multiobjective optimization problem and some effective algorithms are proposed. Compared with mathematical programming, evoutionary algorithms are very suitable to solve multiobjective optimization problems, because they deal simultaneously with a set of solutions and find a number of Pareto optimal solutions in a single run of algorithm. Additionally, they are less susceptible to the shape or continuity of the Pareto front and can approximate to the discontinuous or nonconvex Pareto optimal front.

D. Lei (✉) · Z. Wu
Institute of Automation, Shanghai Jiaotong University,
1954 Huashan Road,
Shanghai, PR China
e-mail: dmlei@mail.sjtu.edu.cn
Tel.: +86-216-2932070
Fax: +86-216-2932070

Van Veldhuizen et al. [2] classified MOEA into three categories in terms of decision-maker (DM)'s preference to the final solutions being made either before, during or after the optimization process: (1) prior preference articulation, DM first combines the differing objectives into a scalar cost functions and then a single objective algorithm is applied to search the final solution. (2) Progressive preference articulation, decision-making and optimization are intertwined. (3) Posterior preference articulation, DM is presented a set of Pareto optimal candidate solutions provided by MOEA before the decision is made. Most of MOEA belong to the third class. Moreover, there exist some common features among the algorithms presented in the past 5 years, For instances, a external population is set up to keep down the nondominated solutions obtained by algorithm, when the actual size of the external population exceeds the predetermined scale, an effective approach is applied to remove some individuals from the external population. The algorithms with the above feature include PAES [3], M-PAES [4], SPEA [5], SPEA 2 [6], NSGA [7], NSGA 2 [8], RDGA [9] etc.

Unlike single objective evolutionary algorithm (SOEA), MOEA must assign fitness for each individual in terms of the Pareto dominance relation among individuals and some individual information including density value. In addition, MOEA basically doesn't choose the reproduction approach based on fitness value, for instance, roulette, and utilize the reproduction based on local competition, where tournament selection is used in genetic algorithm and $\mu+\lambda$ or (μ,λ) selection exploited in evolutionary strategy.

The application of MOEA to job shop scheduling has attracted a few people's attentions. Sridhar et al. [10] developed a multiobjective genetic algorithm (MOGA) to the problem of scheduling in flow shop and cellular manufacturing systems with the objectives of minimizing makespan, total flow time and machine idleness. Ishibuchui et al. [11] presented a MOGA for flow shop scheduling, where the weight coefficients of each objective are stochastically determined and the local search is combined with genetic operators. Ponnambalam et al. [12] proposed a MOGA to derive the optimal machine-wise priority dispatching rules to resolve the conflict among the contending jobs in the Giffler and Thompson (GT) procedure applied for job shop scheduling, the objective is to minimize the weighted sum of makespan, the total idle time of machines and the total tardiness. Kacem et al. [13] presented a combination approach based on the fusion of fuzzy logic and evolutionary algorithm for flexible job shop scheduling. Some progresses on multiobjective job shop scheduling have been made; however, MOEA such as SPEA, which can provide a number of Pareto optimal candidate solutions, is hardly applied to job shop scheduling.

In this paper, crowding-measure-based multiobjective evolutionary algorithm (CMOEA) is designed and applied to job shop scheduling; the objective is to simultaneously minimize makespan and the total tardiness. The rest of the paper is organized as follows. In Sect. 2, several basic concepts of Pareto optimality are introduced. The description on CMOEA is done in Sect. 3 and the discussion on job shop scheduling appears in the next section. The tests on 15 job shop problems are executed respectively for CMOEA, SPEA and the corresponding analyses and comparisons are finished in Sect. 5. The final conclusions occurs in Sect. 6.

2 Pareto optimality

In general, the multiobjective optimization problem is described as follows.

$$\begin{array}{ll} \max & f(x) = [f_1(x), f_2(x), \cdots f_k(x)] \\ Subject\ to & g_i(x) \leq 0 \quad i = 1, 2, \cdots h \quad x \in R^n \end{array} \tag{1}$$

where f_i $1 \leq i \leq k$ is objective function, g_i is constraint, $x \in R^n$ is decision variable.

When faced with only a single objective, an optimal solution is one that maximizes the objective given model constraints; however, more than one objective is required to simultaneously optimized, the optimal solution is often the Pareto optimal set. The several concepts used in MOEA are introduced below.

(1) Pareto dominance, solution x^0 dominate $x^1 (x^0 \succ x^1)$iff $f_i(x^0) \geq f_i(x^1) \quad i = 1, 2, \cdots k \quad f_i(x^0) > f_i(x^1) \quad \exists i \in \{1, 2, \cdots k\}$.
(2) Pareto optimal, solution x^0 is Pareto optimal iff $\neg \exists x^1 : x^1 \succ x^0$.
(3) Pareto optimal set, the set P_S of all Pareto optimal solution, $P_S = \left\{ x^0 \middle| \neg \exists x^1 \succ x^0 \right\}$.
(4) Pareto optimal front or trade-off surface, the set P_F of all objective function vectors corresponding to the solutions in P_S . $P_F = \{f(x) = (f_1(x), f_2(x), \cdots f_k(x)) | x \in P_S\}$.

The size of Pareto optimal set is often infinite; it is obvious that the population with the limited scale cannot obtain the whole set P_S through evolution in the limited generations. To acquire a limited subset representing the whole set P_S is feasible and the final purpose of MOEA. In most MOEA, the external population or the externally stored nondominated solution set are utilized to make the nondominated solutions uniformly distribute on the whole Pareto optimal front.

3 Crowding-measure-based multiobjective evolutionary algorithm

3.1 The analysis on external population

External population P' has been the inseparable part of MOEA. In general, each newly produced nondominated solution is directly added into P', if the size of P' exceeds the predetermined scale of P' after the new solution is inserted, some members of P' must be removed to guarantee the maximum size of P' to be fixed in terms of the following conditions.

(1) Dominated by the new nondominated solutions in P'.
(2) Meet the conditions decided by the individual information such as density of individual.

For instance, in PAES, if the solution increases the diversity of P', it will replace the member of P' located in the most crowded region, while in the adaptive archiving algorithm presented by Knowles et al. [14], if the new nondominated solution is outside the old boundaries of grid or the most crowded grid, it also replaces the one located in the crowded grid. The number of individuals in a grid is a kind of individual information; the distance between different individuals is another type of information. A new individual information is defined as follows.

Definition 1 S is the set of some individuals. The Euclidean distance between the individual i and other members of S in objective space is permuted from small to big, d_i^1 and d_i^2 are the minimum two distances, the crowding measure d_i (in S) of individual i, $d_i = (d_i^1 + d_i^2)/2$.

The crowding measure of individual i reflects the distribution of other individuals around i. The smaller d_i is, the more the number of individuals surrounding i is. Compared with the number of individuals in a grid, the crowding measure exactly describes the relative position relation among different individuals.

The procedure constructing and adjusting external population P' is described as follows.

(1) If the actual size of P' doesn't exceed the predetermined scale, directly insert the nondominated solutions into P'.
(2) If the actual size of P' is equal to the prefixed value, the population P' will be adjusted according to the following principle.
   (a) If the new nondominated solution dominates some members of P', remove these dominated members from P' and assign the new solution into P'.
   (b) If the above Pareto dominance relation doesn't exist, insert the new nondominated solution into P' and remove a member with minimum crowding measure from P'.

When the crowding measure of individual of P' is computed, the individual out of P' doesn't take part in the computation. The above procedure can exactly remove the members concentrating on a location from P' and make the remained solutions evenly distribute on objective space because of the precise description of the crowding measure on the relative position relation among individuals.

3.2 Analysis on fitness assignment

Pareto dominance relation is the foundation of fitness in many MOEA, generally, the individuals are classified into two categories in terms of Pareto dominance relation: dominated and nondominated, and the fitness of the dominated one is always better than that of the nondominated. If there is no Pareto dominance relation, some individual information such as density value is frequently used to determine fitness value. For instance, in SPEA 2, $fit(i)=R_i+D_i$, $fit(i)$ is the fitness of individual i, R_i is raw fitness, the density $D_i=1/(2+\sigma_i^k)$, σ_i^k is the Euclidean distance to the kth nearest individual. While in NSGA 2, the density value of the individual is estimated by the crowding distance, and the crowding distance of an individual is the average side length of the cuboid formed by using its nearest two neighbors as the vertices.

Many MOEA don't pay attention to differ members of P' from the nondominated solutions outside P' on fitness; however, the individuals in P' are supposed to be better than any solutions out of P' because the nondominated solutions outside P' must be the ones removed from P' in terms of the adjustment procedure of external population. In this paper, a novel fitness assignment approach is proposed to guarantee that the fitness value of members of P' is greater than that of any solutions outside P'.

The fitness assignment procedure is shown as follows.

(1) Compute the crowding measure of each individual i in P', $fit(i)=d_i$, any solutions outside P' don't participate in computation.

(2) Compute the crowding measure of each nondominated solution i in the grey region of Fig. 1, $fit(i)=d_i$, Compute the crowding measure of each dominated solution i in the grey region, $fit(i)=\alpha d_i$, $\alpha\leq 0.25$. In the above procedure, when computing the crowding measure of a solution in a gray region; any solutions outside the gray region don't participated in the corresponding computation.

(3) For each dominated solution i in the black region of Fig. 1, compute the minimum distance between i and the member of $P' \; dist_{iP'}=min\{dist_{ij}, j\in P'\}$, $fit(i)=-dist_{iP'}$.

In Fig. 1, the nondominated solutions in P' are sequentially permuted in the objective space in terms of the first objective function value, each member of P' dominates the solutions in the black region (the dominated region), when there are N' members in P', the number of gray region (critical region) will be $N'+1$. When the crowding measure of individuals inside a critical region is computed, vertices of the region corresponding to the solutions in P' are included in the region.

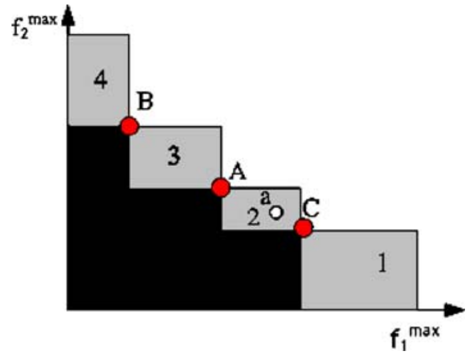

Fig. 1 Regions related to the individuals in P' ● the individual of P'

It is obvious that the solutions inside a critical region possess smaller fitness than the vertices of the region, for instance, the point a in Fig. 1. A is the point with the lowest crowding measure, obviously, the fitness of solutions located in region 2 and 3 is less than that of point A; similarly, the fitness of solutions in region 1 and 4 doesn't exceed the fitness of A, because if a solution in region 1 has a bigger fitness than A, the crowding measure of the solution will exceed A. A should be removed from P' in terms of the above external population adjusting procedure, it is contradictory to the fact. Additionally, the fitness of some dominated solutions is bigger than some nondominated solutions because the number of other individuals around the dominated solution is very small. To give high fitness for these dominated solutions will be beneficial to produce new individuals with small crowding measure.

3.3 The algorithm description

Unlike SPEA 2 and NSGA2, CMOEA first classifies the individuals into four categories in terms of the Pareto dominance relation: the nondominated solutions in P', the nondominated solutions inside the critical region, the dominated solutions inside the critical region, and the dominated solutions inside the dominated region, and then computes the fitness values of each kind of individuals. In addition, the crowding measure is applied to adjust external population and assign fitness for individuals.

The description of CMOEA is recorded as follows.

Step1 Set the parameters of the algorithm, population scale N, the maximum scale N'_{max} of external population P', crossover probability P_c, mutation probability P_m, the maximum generation *gen_max*, let *gen*=1.

Step2 Stochastically generate initial population P and copy the nondominated solutions of P into P'.

Step3 Determine fitness values for each individual in population P and the external population P' and choose the individuals in P+P' into next generation by using binary tournament reproduction.

Step4 Choose individuals into mating pool in terms of crossover probability and then randomly select two individuals from the pool and exchange some genetic bits between them until the mating pool become empty.

Step5 Perform mutation operation for the chosen individuals according to mutation probability.

Step6 Compute the objective vectors corresponding to each individual in P+P' and add the nondominated solutions into P', if the actual size N' of P' meeting $N' > N'_{max}$, adjust external population P' using the approach introduced in Sect. 3.1.

Step7 gen=gen+1, if gen<gen_max, go to step 3, else terminate search.

4 Multiobjective job shop scheduling

The general job shop scheduling considers n jobs to be processed on m machines. The processing of a job on a certain machine is referred to as an operation. The processing time of each operation are fixed and known in advance. Every job passes each machine exactly once in a prescribed sequence. Each job should be delivered before due-date. Makespan and the total tardiness are respectively regarded as objective in many job shop-scheduling models. In this paper, these two objectives are simultaneously minimized.

(1) Makespan: $makespan$=max[C_i]where C_i is the completion time of job i.

(2) Total tardiness of jobs: $\text{Total Tardiness} = \sum_{i=1}^{n} \max[0, L_i]$ where L_i is the lateness of job i.

In this study, Priority rule-based representation is used. The representation makes each chromosome a string containing n×m priority rules; each gene represents a priority rule. Five priority rules such as FCFS, LPT, SPT, CR, and S/OPN are chosen. The priority rule and its corresponding gene value are given as follows, 0——FCFS, 1——LPT, 2——SPT, 3——CR, 4——S/OPN. Each chromosome (p_1,p_2,⋯,p_{nm}) corresponds to a scheduling plan in terms of the following procedure.

The encoding procedure is described as follows.

(1) Let t=1, PS is empty, S_t is the set of operations which can be scheduled in the t iteration.

(2) Determine $c(o^*)=min\{c(o_{ij})|o_{ij} \in S_t\}$ and the corresponding machine m*.

(3) Select a operation o_{im^*} from the conflict set C_t= $\{o_{im^*} \in S_t | \sigma_{im^*} < c(o^*)\}$using priority rule p_t, $PS_{t+1}=PS_t \cup \{o_{im^*}\}$, delete o_{im^*} from S_t, add the next operation of job i into S_t and form S_{t+1}.

(4) t=t+1, repeat step 2, 3, and 4 until a complete scheduling plan is obtained.

o_{ij} denotes the operation of job i processed on machine j. $c(o_{ij})$, σ_{ij} respectively indicates the earliest completion time and the earliest beginning time of operation o_{ij}.

5 Computational results

Fifteen job shop scheduling problems are respectively solved by CMOEA, SPEA to test the capability of CMOEA on scheduling. Ponnambalam et al. [12] provided the due date data for ft06. For problems with 10 jobs, the due date of job 2,3 is 1.5 times the total processing time of the corresponding job, the deadline of job 10 is equal to its total processing time and the due date of other jobs is 2 times the corresponding total processing time. For a problem with 20 jobs, the due date of job 2,3,11 is 1.5 times the corresponding total processing time, the deadline of job 20 is equal to its total processing time and the due date of other jobs is 2 times the corresponding total processing time.

In this study, $\tilde{C}$ metric is used to compare the approximate Pareto optimal set respectively obtained by SPEA

Table 1 The objective function vectors of the best solutions provided by two algorithms

Problem	SPEA	CMEA	Problem	SPEA	CMEA	Problem	SPEA	CMEA
FT06 6×6	58,4	58,4	ABZ7 20×15	790,487	786,350	ORB4 10×10	1130,709.5	1138,314.5
	60,1	60,1		783,396	789,465		1148,616.5	1139,690.5
	62,0	62,0		794,354	792,353		1168,483.5	1154,569.5
Ft10 10×10	1059,180	1057,274	ABZ8 20×15	805,707.5	817,293	ORB5 10×10	1002,1	988,23
	1076,196	1072,187		808,624,5	819,552.5		1012,5	989,45
	1093,267	1085,156		810,582,5	824,251			994,18
Ft20 20×5	1269,6629	1283,6858	ORB1 10×10	1160,718	1160,858	LA 26 20×10	1405,4436	1366,3539
	1276,6928	1287,6668		1193,413	1161,393.5		1428,4346	1375,3537
	1287,6376	1306,6644		1191,469	1188,381.5			1394,4063
ABZ5 10×10	1306,439	1277,422	ORB2 10×10	949,192	941,36	LA 27 20×10	1451,2960.5	1451,2968.5
	1316,486	1296,360		952,190	952,41		1452,2512.5	1452,2611.5
				956,130	956,31			
ABZ6 10×10	981,212	979,348	ORB3 10×10	1164,890	1167,367	LA28 20×10	1400,3111	1398,3553
	994,216	988,155		1165,793	1173,364		1414,2926	1410,3339
	1002,166	993,309		1158,799	1191,596			

Table 2 The comparison results between SPEA and CMEA

Problem	$\tilde{C}(C,S)$	$\tilde{C}(S,C)$	Problem	$\tilde{C}(C,S)$	$\tilde{C}(S,C)$	Problem	$\tilde{C}(C,S)$	$\tilde{C}(S,C)$
FT06	0.095	0.221	ABZ7	0.807	0.821	ORB4	0.925	0.692
Ft10	0.894	0.6956	ABZ8	0.787	0.805	ORB5	1	0.529
Ft20	0.611	0.409	ORB1	0.889	0.667	LA 26	1	0.243
ABZ5	1	0.636	ORB2	1	0.642	LA27	0.846	0.769
ABZ6	0.837	0.837	ORB3	0.8	0.5	LA28	0.789	0.806

and CMOEA. $\tilde{C}(A,B)$ measures the fractions of members of B that are dominated by members of A.

$$\tilde{C}(A,B) = \frac{|\{b \in B : \exists a \in A, a \succ b\}|}{|B|} \quad (2)$$

The parameters of two algorithms are set as follows. N=80, N'_{max}=20, P_c=0.85, P_m=0.1, for problem LA26, LA27, LA28, *gen_max*=150, for other problems, *gen_max*=100. For SPEA and CMOEA, two points crossover and two points uniform mutation are chosen. When the chosen mutation is performed, first some individuals are selected in terms of P_m, randomly pick out two gene bits from the chosen individual and then replace them with new values uniformly selected from all possible values of each bit. For each problem, each algorithm stochastically runs 30 times. The final computational results are shown in Table 1. There are three groups of data in the unit grid of the column 2,3,5,6,8, and 9, which are objective function vectors of the best solutions. Table 2 summarizes the comparison results between SPEA and CMOEA in terms of $\tilde{C}$. metric. $\tilde{C}(C,S)\left(\tilde{C}(S,C)\right)$ indicates the fraction of all solutions in P' obtained by SPEA (CMOEA) in 30 runs that are dominated by the ones (in P') of CMOEA (SPEA) in all runs.

As shown in Table 1, both CMOEA and SPEA provided a group of solutions with ideal objective function values. The first objective of these solutions is just about 120 bigger than the optimal makespan of the problems and some makespan values are very close to the optimal results; while the second objective of these solutions is reasonable and acceptable.

It can be concluded from Table 2 that CMOEA outperforms SPEA in most cases. CMOEA obtains lower $\tilde{C}$ metric than SPEA on ten problems of 15. Moreover, the difference of $\tilde{C}$ between two algorithms is significant, for instance, for problem ORB3, $\tilde{C}(S,C) - \tilde{C}(C,S) = 0.3$. On the contrary, SPEA just slightly outperforms CMOEA on four problems of 15 and the difference on $\tilde{C}$ between two algorithms is trivial.

In this study, the differences between CMOEA and SPEA just include the external population adjustment approach and fitness assignment procedure. These differences make CMOEA have preference to SPEA on job shop scheduling. For instance, CMOEA assigns a different fitness for individuals while many different solutions of SPEA are assigned the same fitness; as a result, CMOEA possesses a greater possibility than SPEA in eliminating the individuals with lower fitness from population in a generation and promotes the search efficiency.

6 Conclusion

Multiobjective evolutionary algorithm has attracted much attention in the past decade. However, the application of MOEA to practical problems such as job shop scheduling is seldom considered. In this study, crowding-measure-based multiobjective evolutionary algorithm (CMOEA) is first designed, and the new algorithm is applied to job shop scheduling to simultaneously minimize the makespan and the total tardiness of jobs. The comparison between CMOEA and SPEA in solving 15 scheduling problems demonstrates that CMOEA is very suitable to job shop scheduling.

References

1. Schaffer JD (1985) Multiple objective optimization with vector evaluated genetic algorithms. In: Proc lst international conference on genetic algorithm, Lawrence Erlbaum Associates, Hillsdale, pp 93–100
2. Van Veldhuizen DA, Lamont GB (2000) Multiobjective evolutionary algorithms: analyzing the state-of-the-art. Evol Comput 8(2):125–147
3. Knowles JD, Corne DW (2000) Approximating the nondominated front using the Pareto archive evolutionary strategy. Evol Comput 8(2):149–172
4. Knowles JD, Corne DW (2000) M-PAES: a memetic algorithm for multiobjective optimization. In: Proceedings of the IEEE Conference on Evolutionary Computation, ICEC 1:325–332
5. Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Trans Evol Comput 3(4):257–271
6. Zitzler E, Laumanns M, Thiele L (2001) SPEA2: improving the strength Pareto evolutionary algorithm. Swiss Federal Institute of Technology, Lausanne, Switzerland, pp 103 (Tech. Rep. TIK-Rep)
7. Srinivas N, Deb K (1994) Multiobjective optimization using nondominated sorting in genetic algorithms. Evol Comput 2 (3):221–248
8. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithms: NSGA. IEEE Trans Evol Comput 6(2):182–197
9. Lu H, Yen GG (2003) Rank-density-based multiobjective genetic algorithm and benchmark test function study. IEEE Trans Evol Comput 7(4):325–343

10. Sridhar J, Rajendran CS (1996) Scheduling in flow shop and cellular manufacturing with multiple objectives a genetic algorithm approach. Prod Plan Control 7(4):374–382
11. Ishibuchi H, Murata T (1998) A multi-objective genetic local search algorithm and its application to flow shop scheduling. IEEE Trans Syst, Man Cybern: Part C 28(3):392–403
12. Ponnambalam SG, Ramkumar V, Jawahar N (2001) A multi-objective genetic algorithm for job shop scheduling. Prod Plan Control 12(8):764–774
13. Kacem I, Hammadi S, Borne P (2002) Pareto-optimality approach for flexible job shop scheduling problems: hybridization of evolutionary algorithm and fuzzy logic. Math Comput Simul 60:245−276
14. Knowles J, Corne D (2003) Properties of an adapative archiving algorithm for storing nondominated vectors. IEEE Trans Evol Comput 7(2):100–116