Towards KAB\textsuperscript{2}S: Learning Key Knowledge from Single-Objective Problems to Multi-Objective Problem

Xu Wendi\textsuperscript{1,2}, Wang Xianpeng\textsuperscript{1,2}, Guo Qingxin\textsuperscript{1,2}, Song Xiangman\textsuperscript{1,2}, Zhao Ren\textsuperscript{1,2}, Zhao Guodong\textsuperscript{1,2}, Yang Yang\textsuperscript{1,2}, Xu Te\textsuperscript{1,2}, He Dakuo\textsuperscript{1}

\textsuperscript{1} College of Information Science and Engineering, Northeastern University
\textsuperscript{2} Key Laboratory of Data Analytics and Optimization for Smart Industry, Ministry of Education, China
wangxianpeng@ise.neu.edu.cn

\textbf{Abstract}—As “a new frontier in evolutionary computation research”, evolutionary transfer optimization (ETO) will overcome the traditional paradigm of zero reuse of related experience and knowledge from solved past problems in researches of evolutionary computation. In scheduling applications via ETO, a quite appealing and highly competitive framework “meeting” between them could be formed for both intelligent scheduling and green scheduling, especially for international pledge of “carbon neutrality” from China.

To the best of our knowledge, our paper on scheduling here, serves as the 1\textsuperscript{st} work of a class of ETO frameworks when multi-objective optimization problem “meets” single-objective optimization problems in discrete case (not multitasking optimization). More specifically, key knowledge conveyed for industrial applications, like positional building blocks with genetic algorithm based settings, could be used via the new core transfer mechanism and learning techniques for permutation flow shop scheduling problem (PFSP).

Extensive studies on well-studied benchmarks validate firm effectiveness and great universality of our proposed ETO-PFSP framework empirically. Our investigations (1) enrich the ETO frameworks, (2) contribute to the classical and fundamental theory of building block for genetic algorithms and memetic algorithms, and (3) lead towards the paradigm shift of evolutionary scheduling via learning by proposal and practice of paradigm of “knowledge and building-block based scheduling” (KAB\textsuperscript{2}S) for “industrial intelligence” in China.

\textbf{Keywords}—evolutionary transfer optimization, green scheduling, transfer learning, data analytics, system optimization, carbon neutrality

1. Introduction

“\textbf{Artificial Intelligence} (AI) is a science and a set of computational technologies that are inspired by—but typically operate quite differently from—the ways people use their nervous systems and bodies to sense, learn, reason, and take action”\cite{23}. That is what AI \cite{1,2,22,23,42,43,52} is stated in the 2016 report of “One Hundred Year Study on Artificial Intelligence (AI100)” (held by Stanford University). When narrowing \textbf{Artificial Intelligence} for industries within context of China, AI is formulated as \textbf{Industrial Intelligence} (II)\cite{1,2}.

\textbf{Industrial Intelligence} has great importance in China’s industrial upgrading and transformation into a true industrial power.”\cite{2} Towards industrial intelligence, knowledge\cite{42,52} acts as both “the core production factor” \cite{52} and “a new mode of innovation”\cite{52} for systems and algorithms in smart manufacturing (MS). A specific representation of knowledge in evolutionary scheduling, especially solved by genetic algorithm (GA) and GA based memetic algorithm (MA)\cite{25}, is building block (BB), which is regarded as the “holy grail to GA optimization”\cite{8,24}. Therefore, we emphasize the core component in \textbf{Industrial Intelligence} for China’s industrial upgrading and transformation: \textbf{knowledge}.

Those industrial \textbf{knowledge} of both specific (like BB) and general forms, structured and unstructured representation, explicit and implicit features, could be mined and learned during the decision-making process which allocates related resources to operational tasks in given time periods towards the optimization of one or more objectives in production management. Recently, due to the growth of algorithmic toolkit for mining and learning, for scheduling applications in Industrial Intelligence, we are fortunate to witness the paradigm shift of evolutionary scheduling via learning\cite{1,2,29}, which is abstracted as a paradigm of “knowledge and building-block based scheduling” (KAB\textsuperscript{2}S) here.

For KAB\textsuperscript{2}S, the paradigm contains both intra-task learning and inter-task learning. The latter includes the new tool of evolutionary transfer optimization (ETO), with learning/transferring ability across problems/tasks, which will be investigated here. So this \textbf{paper} on ETO based scheduling is the \textbf{first} in the planned \textbf{series} to be issued as a pioneer part of the “paradigm shift”\cite{19} of KAB\textsuperscript{2}S.

Firstly, let’s begin with “ETO meeting scheduling”, “Relation is ubiquitous, whereas, isolation is seldom”\cite{49}, which is the philosophy behind ETO, “a paradigm that integrates EA solvers with knowledge learning and transfer across related domains to achieve better optimization efficiency and performance”\cite{3}. ETO, has served as a new frontier in evolutionary computation, which has internal connections with extreme learning machine\cite{34}, transfer learning\cite{5,50}, deep learning \cite{4,5,15,52} and evolutionary algorithms\cite{26,45}. In manufacturing and services systems, scheduling acts as a decision-making process which allocates resources to tasks during given periods for optimization of one or more objectives. We hope to investigate the new and relatively rare studied solution framework(s) of “ETO...
meeting scheduling”, especially for scheduling science and technologies in system engineering or production management[2] towards green [2] [16] and intelligent scheduling[2].

Then, “ETO meeting scheduling” for Complex Optimization(single-objective to multi-objective optimization). According to the types of problems, ETO can be classified as follows[3]: “(1) ETO for optimization in uncertain environment, (2) ETO for multi-task optimization(MTO) or multi-factorial optimization(MFO), (3) ETO for multi-many objective optimization, (4) ETO for machine learning applications, (5) ETO for complex optimization”[3,49]. When ETO meets scheduling for complex optimization particularly, we can narrow the setup as a case from single-objective problems optimization to multi-objective problem optimization[10]. Since single-objective problems(SOPs) are relatively simpler and related problems “for transferring knowledge into its corresponding multi-objective problem(MOP), which will greatly reduce computational complexity of scheduling, like permutation flow shop scheduling(PFSP), that often is a NP-hard problem”[10, 49].

Lastly, learning/transferring specific key knowledge, that is, building blocks between PFSPs. Building blocks serves as the main objects/roles that are manipulated implicitly or explicitly for the successful performance of GAs and MAs[7][8]. The exploitation of BBs in GAs and MAs, especially the tight ones, are regarded as the “holy grail to GA optimization”[8]. (From now on, “BB” and “block” are exchangeable in the paper.) The definition for BBs or genetic linkage is intertwined tightly with recombination operator [10] in both biological and artificial systems. “If the genetic linkage between these genes is tight, the crossover operator disrupts them with a low probability and transfers them all together to the child individual with a high probability. On the other hand, if the genetic linkage between these genes is loose, the crossover operator disrupts them with a high probability and transfers them all together to the child individual with a low probability.”[2] Mathematically, dependencies among decision variables, like elements in the vectors, indicate the building blocks[12]. From a more biologically plausible perspective, we got 3 levels structure, gene→BB→chromosome. The BBs for machine scheduling in production management, especially PFSP, has been already identified by [9][13]. That is, engineering practice has proved the existence of BBs and the successful manipulation of BBs via non-learning tools[9]. However, the manipulation of BBs via learning tools[13] is relatively rare( not only in cases of machine/shop scheduling problems, but also in other well-known combinatorial problems like vehicle routing problems, travel salesman problems and so on). Therefore, we recommend BBs as key knowledge for ETO and learning in PFSP.

What should be highlighted here is that 1st and 2nd contributions are almost the same as in our conference paper[49] (finding of “first work of ETO for PFSP” is not figured out in conference paper), 3nd and 4th contributions(there are only 3 contributions in that paper) in this paper are totally different from those in that paper, because of additional empirical test and depth insight of classical building block theory.

2. A step towards carbon neutrality and green scheduling via a devotion to avoid scheduling operations from scratch.

3. An extensive study of different combination of operators show the functional parts of our SMO framework.

4. We further extend the classical building block theory within new learning settings to enrich the fundamental GA and MA theory.

2. Related works

The most related paper to ours is [10]. To the best of their knowledge, their work serves as the first work to boost evolutionary multi-objective optimization via knowledge transferred from corresponding single-objective problems. And their problems are continuous, not discrete. For example, quadratic assignment problem(QAP), linear ordering problem (LOP), travel salesman problem(TSP), job shop scheduling problem(JSP) are discrete. In our work, we roughly follow the same framework in [10], such as transferring experience by direct injection of external populations from source task of all single objective problems to corresponding multi-objective problem(target task) “every G generations, where G is the gap”[49]. Fundamental dissimilarities between continuous and discrete cases prevents us from directly following their deployment of “mapping/connection between populations in source task and target task via denoising autoencoder”[49]. The mapping lacks physical meaning for permutation representation in case of PFSP. Instead, that mapping or connection is constructed in our work as follow. We add the design of intra-task learning via clustering which is not deployed in their work to select potential elite solutions with potential good positional BBs, then directly inject the external population(which may including those elite solutions) from source task to target one.

In the following, another 4 important related works are given. First is ETO for MTO in discrete case of TSP, QAP, LOP, and JSP. Multifactorial evolutionary algorithm was designed to explore the power of evolutionary multitasking, which can act as the engine for simultaneous optimization of multiple permutation-based combinatorial problems in supply chain networks[11]. In order to “achieve the adaptive ability, a unified representation design and selection operator are applied”[49]. Secondly, ETO for MTO in discrete case of routing. In the work [12], a memetic computing framework is developed, which can evolve and learn knowledge meme which traverses 2 different but related domains for enhancement of evolutionary search performance. For discrete case, a realization is implemented on two NP-hard vehicle routing domains, that is, capacitated arc routing problem and capacitated vehicle routing problem. Thirdly, “machine learning based intelligent optimization(MA) for PFSP”[49]. In [13], machine learning based MA is designed for PFSP in the setting of multi-objective. The algorithm is called ML-MOMA, one of whose main development is design of local search operator via machine learning. In ML-MOMA, historical data during the search optimization of PFSP is fully

![Fig. 1. A simple example of building blocks in PFSP [9].](Image)
utilized. And the clustering method is also introduced to selected better permutation solutions that are representative ones for refinement in local search. Duplicated searches thus are effectively avoided on similar individual permutation solutions. Besides MA above, another method in the whole family of intelligent optimization, like simulated annealing (SA), can also tackle PFSP as follows. Lastly, "residual learning based intelligent optimization (SA) for PFSP"[49]. To solve the optimization of PFSP, [14] introduces an improved method of SA which are armed with residual learning setting. The neighborhood are defined in PFSP. And "the trained/ learned/fitted parameters are further stored in the SA for greater search efficiency"[14,49].

3. Background: transferred knowledge, science clustering, BB theory and test problem

3.1. Transferred knowledge: positional BB dominates other BBs

Makespan(Cmax), the minimum result of the maximum completion time, plays the role of the most powerful and core optimization objective/driving force among management objectives in many-objective, multi-objective and single-objective machine scheduling problems. Cmax dominates total flow time(TFT), maximum lateness, total tardiness and so on. In other words, it should be put forward firstly, better Cmax usually is also strongly correlated with other better optimization objectives. For combinational problems, especially for machine scheduling or PFSP, those building blocks remains unclear, and some important clues may help us. Positional, precedent and adjacent types of information units in combinational problems are believed to exit. In TSP, positional structures are dominated by adjacency structures, while positional structures matters more than adjacency structures for optimization objective of Cmax in case of PFSP[27]. Based on those observations above, we tend to believe that there may exit three kinds of BB, that is, positional, precedent and adjacent BBs. In later experiments, we will focus on the 1st one, positional BB. Therefore, we could summarize that BBs of positional type help improve Cmax, and also improve other objectives inherently, the significance of positional ones is highlighted here.

3.2. Science clustering: mining positional BB via unsupervised learning

Clustering is one of unsupervised learning methods to classify elements into categories according to their similarity measure. Its applications range from biology to astronomy, pattern recognition, and so on. In [6], “science clustering” (because it was published at science, so we named it as “science clustering”) is based on the deep observation that centers of clusters in sample space are characterized by both “a relatively higher density than points in their neighborhoods”[6, 49] and “a relatively long distance from points that have higher densities” [6, 49]. For PFSP here, we implement science clustering via hamming distance metric, which is widely accepted and used in evolutionary computation research. To our surprise, hamming distance also works from opposite sides to the same characterization”[6, 49].

3.3. BB theory: Goldberg’s decomposition theory (7 steps) and BB processing pipeline(5 steps)

The BB theory/hypothesis is mainly based in Goldberg’s decomposition theory, which contains 7 steps as follow.

1. “Know what GAs process” [21]—BBs.
2. Solve optimization problems which have bounded BB difficulty.
3. Ensure that the supply of raw BBs are adequate.
4. Ensure that the “market share” [21] of superior BBs increases.
5. Know “BB takeover” [21] and the models of convergence times.
6. Ensure that GAs make the BB decisions well.
7. Ensure that the mixing of BBs is good.

Based on the decomposition theory above, they propose the BB processing pipeline with 5 steps: (1)creation, which creates the raw supply of BBs,(2)identification, that attempts to identify the good BBs,(3)separation, which separates superior BBs,(4)preservation, which maintains the good BBs, and (5)mixing, that reassembles good BBs [21], characterization.

3.4. Test problem: PFSP

The scheduling problem in permutation flow shop is named PFSP. Therefore, PFSP[33, 35, 38] is formulated as follows: “each job/operation is to be processed sequentially on machines, given their processing time of the operation”[49]. Every machine can process one operation or job at most, and every job can be processed at most on one machine during optimization. The sequence of job permutation is the same on every machine. Optimality[32] of solution/search space is due to the satisfaction of the objectives. As to objectives in PFSP, we chose makespan and total flow time as optimization goals towards optimality.

4. The framework across tasks: ETO_PFSP

4.1. 4 frameworks: SOO , MOO, MFO, and SMO

As in the paper[44] of multitasking in ETO, there are 3 kinds of optimization problems, single-objective optimization (SOO), multi-objective optimization (MOO) and multi-factorial optimization (MFO). So the framework of ETO_PFSP in this paper is abstracted as single-objective to multi-objective /many objective optimization(SMO). We firstly propose the definition of SMO. Both frameworks of SOO and MOO are not new, whereas, both MFO and SMO belonging to ETO are new relatively. The published time in June, 2016 of the MFO work[44] may be remarked as the born time of MFO, and the time of October, 2020 when the paper[10] is published may be viewed as the born of SMO for continuous case. So SMO is more newer than MFO. In the framework of MFO, it focus on solve problems simultaneously like multitasking in machine learning, while SMO is inclined to the target task, like transfer learning in machine learning. Our ETO_PFSP is a combinatorial case of SMO, or SMO for PFSP.

4.2. 5 bags, 5*2 groups, 5*2*4* tasks: e.g. bag 1 : group 1, t1_wc(t_1, w_1), t2_wc and t2e_wc; group 2, t1_nc(t_1, n_1, t_1), t2_nc and t2e_nc ...

Overview (in Figure 2). In ETO_PFSP, for each bag, we set two task groups.
Bag 0 is full of group 1 and group 2. “Group 1 owns 4 tasks, namely, t1_wc including two sub tasks(t_wc 1.0, t_wc 1.1), t2_wc and t2e_wc, where “wc” means with clustering and “e” is external transferring from t1_wc, sharing the same tool kit of W-X-L (only probabilities vary in X, more is in V.) All above is the same for group 2 of t1_nc(t nc 1.0, t nc 1.1), t2 nc and t2e nc, except that no clustering (named “nc”) is in W”[49]. In bag 0, each case calculates the measure of hamming distance in the job permutation from the 15th one to last job (just test the special distribution of positional BB).

Bag 1 stores group 3 and group 4. And also, MWT modifications are set (3 modifications of “MWT” are added: (1) just remove M(M) in X, (2) study the whole(W) chromosome, calculate hamming distance from the first job to last job instead of from the 15th job to last job in bag 0, (3) task 1.0 choose TFT(T) as the objective for local search, while all tasks in bag 0 are set with Cmax for local search.) Then just remove L from group 1 and group 2 in bag 0 to achieve group 3 and group 4 for bag 1. Then, Bag 2 stores group 5 and group 6. And also, MWT modifications are set. Moreover, just remove L from group 1 and group 2 in bag 0 to achieve group 5 and group 6 for bag 2. And, Bag 3 is occupied by group 7 and group 8. And also, MWT modifications are set. Just remove X from group 1 and group 2. At last, Bag 4 is occupied by group 9 and group 10. And also, MWT modifications are set. Just remove S from group 1 and group 2.

W-X-L deploys a special operator to choose parents(W), a crossover(X) operator and a local search(L) operator. It’s worth to mention that the family of tasks above shares the same initial(I) population (random) for fair comparison. The phase of selection(S) differs. For S, we use NSGA II, some sorting methods by Cmax objective or TFT objective and so on. Therefore, so many shared parts above from both problem and algorithm sides are elaborately constituted towards a harmony test bed for a well-defined F3.

Details of W (seen in Figure 3). For W, with C and S, (in Figure 4), we choose parents P'. The key component, density peak based clustering (DPBC)[6] is developed in C.

Details of XL (seen in Figure 5), especially X. And more of X is presented in Figure 5. L is just an ordinary insertion operator (both operators are general, e.g., L is just insertion, thus constituting the great generality of the framework).

Details of SS in S (seen in Figure 6). Both “t1_wc/nc and t2_wc/nc have no P0, only t2e_wc/nc needs P0 every G generation”[49]. Both t1_wc/nc 1.0 and 1.1 establish the selection pressure via settings of single objectives, and both t2_wc/nc and t2e_wc/nc use both objectives via NSGA II.
5. Experimental studies and comparisons

5.1. Experimental setup

To test the validation of the framework of ETO_PFSP, we carry out extensive simulation on some PFSP instances in well-studied benchmarks, that is, tai01(20x5), tai42 (50x10) and VFR100_20_1(100x20), e.g., the symbol of 20x5, denotes 20 jobs and 5 machines.

And in our simulation, ETO_PFSP is coded by Python 3.8.8 (3.7.0 also OK), and is run on servers.

The following parameters of ETO_PFSP are set: “N is 100, number of generation is 100. In t1_wc/nc 1.0 and 1.1, [p \times 1, p \times 2/m] are [0.3, 0.7] and [0.1,0.9], respectively. For t2_wc/nc and t2e_wc/nc, it’s [0.2,0.8]. For reference points, tai01 takes range (2500, 1000) to normalize Cmax, and (25000,10000) to normalize TFT; tai42 uses (4200,2500) and (120000, 80000); and instance 3 picks (10000,5000) and (550000, 350000). The gap G is 2. The base-line size of P is 50, modified by a factor K1. For P1, the base-line size is 20+H. And 20 is also adapted by K2, H may be 0, 1, 2 or 3, depending on the solutions with equal distance at cutting distance” [49].

Varying the setup of \([K1, K2]\) from vectors of \([1,0.6], [0.6,0.6]\) to \([1,1]\) in each instance, we get 9 cases (in Figure 7,8,9,10,11), and each case owns total 20 independent runs. In each run, we perform simulation of 8 tasks, that is, 2 task groups in 4.2.

5.2. Simulations and comparisons

In every case, both t2_wc and t2e_wc work with clustering methods, and both t2_nc and t2e_nc are performed without clustering.

In every case, we evolve overall 5(bags) *100 (generations per task) *4 (4 tasks in each group) *2 (wc/nc, that is wc or nc) *20(independent runs) = 80 000 generations!

Furthermore, the whole size of solution space or search space is a factorial (for example, 20x5, means 20! solutions), imposing a tremendous computational challenge to get the 5(bags)*9 results or cases (for cases 7,8 and 9 in bag 0, each case takes a long time of 30+ hours even on a server).
Fig. 7. Cases 1, 2, ..., 9 in bag 0 are shown above, respectively. Two notes:
(1) “stat” means the specified statistics of hypervolume. (2) t2_nc is the baseline of NSGA II, that is, the baseline without both clustering and transferring.

For bag 0, between t2_wc and t2e_wc pair, there is always positive transferring in figure 7(a “big” figure containing 9 results), which tends to validate the successful part of effectiveness(e) in our framework of ETO_PFSP or SMO.

Whereas, ineffectiveness exists in the comparison between t2_nc and t2e_nc.

Those (actually reported in our conference paper[49]) can be seen in detail in the table below (“e” means effectiveness, “ie”, is ineffectiveness, “ee”, denotes great effectiveness, “=”, represents equal).

| Bag 0 | Operators and Transfer Effectiveness |
|-------|-------------------------------------|
|       | Operators                           | t2_wc VS t2e_wc | t2_nc VS t2e_nc |
|       | Notes: g means group                 | t2_wc VS t2e_wc | t2_nc VS t2e_nc |
|       | ee means great effectiveness         |                   |                 |
|       | ie means ineffectiveness             |                   |                 |
|       | ee means effectiveness of clustering |                   |                 |
| case 1| WXLS (g1) XLS (g2), no MWT           | e                 | ie              |
| case 2| WXLS (g1) XLS (g2), no MWT           | e                 | ie              |
| case 3| WXLS (g1) XLS (g2), no MWT           | e                 | ie              |
| case 4| WXLS (g1) XLS (g2), no MWT           | ee               | =               |
| case 5| WXLS (g1) XLS (g2), no MWT           | ee               | =               |
| case 6| WXLS (g1) XLS (g2), no MWT           | ee               | =               |
| case 7| WXLS (g1) XLS (g2), no MWT           | ee               | =               |
| case 8| WXLS (g1) XLS (g2), no MWT           | e                 | ie              |
| case 9| WXLS (g1) XLS (g2), no MWT           | ee               | ie              |

Summary: bag 0, 5 ee + 4 e = e, 6 ie + 3 = = ie

Then, in bag 1,2,3 and 4, we will give a more systematic study of each operator.
For bag 1, between $t_{2\text{wc}}$ and $t_{2e\text{wc}}$ pair, there is almost always (except case 8, positive, not obvious enough) obvious positive transferring effectiveness in figure 8(a figure containing 9 results), which tends to validate that there exists great effectiveness (ee) part in our ETO_PFSP or SMO framework.

Whereas, as to $t_{2\text{nc}}$ and $t_{2e\text{nc}}$, both great effectiveness and normal effectiveness (e) exist.

Those two observations of two pairs of tasks can be summed up in the table II below.

**TABLE II. COMPARISONS IN BAG 1 FOR ETO_PFSP FRAMEWORK**

| Bag 1 | Operators | Transfer Effectiveness |
|-------|-----------|------------------------|
|       | Notes: g means group e means effectiveness ee means great effectiveness ie means ineffectiveness ec means effectiveness of clustering | $t_{2\text{wc}}$ VS $t_{2e\text{wc}}$ VS $t_{2\text{nc}}$ VS $t_{2e\text{nc}}$ |
| case 1 | WXLS (g3) XLS (g4), +MWT | ee ee |
| case 2 | WXLS (g3) XLS (g4), +MWT | ee ee |
| case 3 | WXLS (g3) XLS (g4), +MWT | ee ee |
| case 4 | WXLS (g3) XLS (g4), +MWT | ee e |
| case 5 | WXLS (g3) XLS (g4), +MWT | ee ee |
| case 6 | WXLS (g3) XLS (g4), +MWT | ee e |
| case 7 | WXLS (g3) XLS (g4), +MWT | ee e |
| case 8 | WXLS (g3) XLS (g4), +MWT | e e |
| case 9 | WXLS (g3) XLS (g4), +MWT | ee e |

Summary: bag 1, 8 ee + 1 e = ee, 4 ee + 5 e = e

Then, in bag 2 below, we will focus on the systems based on bag 1, but without local search or memes. Here is another 9 cases as follows.
Fig. 9. Cases 1, 2, ..., 9 are shown above, respectively. Notes: (1) “stat” means statistics of hypervolume. (2) Actually, t2 nc is the baseline NSGA II, without both clustering and transferring.

As to bag 2, between the pair of t2 wc and t2e wc, there is definitely obvious positive transferring effectiveness in figure 9 above, which tends to validate a great effectiveness (ee) part in our framework.

Whereas, for t2 nc and t2e nc, both normal effectiveness (e) and ineffectiveness (ie) can be seen.

Those two observations are summed up below in the table III.

TABLE III.

| Bag   | Operators and Transfer Effectiveness |
|-------|--------------------------------------|
|       | Operators                               |
|       | Notes: g means group                    |
|       | ee means great effectiveness            |
|       | ie means ineffectiveness                |
|       | ec means effectiveness of clustering    |
| case 1| WX S (g5) X S (g6), +MWT               |
| case 2| WX S (g5) X S (g6), +MWT               |
| case 3| WX S (g5) X S (g6), +MWT               |
| case 4| WX S (g5) X S (g6), +MWT               |
| case 5| WX S (g5) X S (g6), +MWT               |
| case 6| WX S (g5) X S (g6), +MWT               |
| case 7| WX S (g5) X S (g6), +MWT               |
| case 8| WX S (g5) X S (g6), +MWT               |
| case 9| WX S (g5) X S (g6), +MWT               |
|       | Summary: bag 2, 9 ee = ee, 4 ee + 5 ie = ie |

Then, let us move forward to bag 3 below, we will abandon the artificial evolutionary sub-systems of genetic algorithm, and just study the local search tools/sub-systems.
Fig. 10. Cases 1, 2, ..., 9 are shown above, respectively. Notes: (1) “stat” means statistics of hypervolume. (2) Actually, $t_2_{nc}$ is the baseline NSGA II, without both clustering and transferring.

And in bag 3, no matter between the tasks of $t_2_{wc}$ and $t_2e_{wc}$, or between the pair of $t_2_{nc}$ and $t_2e_{nc}$, obvious positive transferring effectiveness exist in figure 10, which tends to strongly validate the great effectiveness ($ee$) in SMO.

Those observations are again summed up via the table IV.

| Case | Operators and Transfer Effectiveness |
|------|--------------------------------------|
| 1    | W LS (g7) LS (g8), +MWT ee ee       |
| 2    | W LS (g7) LS (g8), +MWT ee ee       |
| 3    | W LS (g7) LS (g8), +MWT ee ee       |
| 4    | W LS (g7) LS (g8), +MWT ee ee       |
| 5    | W LS (g7) LS (g8), +MWT ee ee       |
| 6    | W LS (g7) LS (g8), +MWT ee ee       |
| 7    | W LS (g7) LS (g8), +MWT ee ee       |
| 8    | W LS (g7) LS (g8), +MWT ee ee       |
| 9    | W LS (g7) LS (g8), +MWT ee ee       |

Summary: bag 3, 9 $ee = ee$, 9 $ee = ee$

Then, let us go to bag 4, we will explore the W operator.
Fig. 11. Cases 1, 2, ..., 9 are shown above, respectively. Notes: (1) “stat” means statistics of hypervolume. (2) Actually, t2_nc is the baseline NSGA II without both clustering and transferring.

Bag 4 tells us that between t2_wc and t2e_wc, and the pair of t2_nc and t2e_nc, you can see only some rules in SMO. We define “ec” as effectiveness of the selection function of clustering. The ec is quite obvious.

The report of observations above are again summed up by the table V.

TABLE V. COMPARISONS IN BAG 1 FOR ETO_PFSP FRAMEWORK

| Case | Operators and Transfer | Effectiveness |
|------|------------------------|---------------|
| Case 1 | WXL (g9) XL (g0), +MWT | ee            |
| Case 2 | WXL (g9) XL (g0), +MWT | e             |
| Case 3 | WXL (g9) XL (g0), +MWT | e             |
| Case 4 | WXL (g9) XL (g0), +MWT | e             |
| Case 5 | WXL (g9) XL (g0), +MWT | e             |
| Case 6 | WXL (g9) XL (g0), +MWT | e             |
| Case 7 | WXL (g9) XL (g0), +MWT | =             |
| Case 8 | WXL (g9) XL (g0), +MWT | =             |
| Case 9 | WXL (g9) XL (g0), +MWT | =             |

Summary: ec.

6. Discussion

6.1. Transferring and learning between 5 bags: 1 table, 4 additional findings

We can summarize Bags 0, 1, 2, 3, and 4 as follow.

TABLE VI. COMPARISONS IN ETO_PFSP FRAMEWORK.
### 6.2. Knowledge of three kinds: general/nurtue and specific/nature balance

From the table, we tend to conclude that transfer effectiveness of the SMO framework is always quite obvious (Bag 4 is aimed to just test selection, not test transferring), especially under clustering conditions(t2_wc VS t2_we). What are reasons for the success? Partly due to the similarities between two landscapes derived by those operators and domains. That is the first finding.

Additional 3 findings are that (1) Comparing Bags 1, 2, and 3, we tend to believe that collaboration between X(GA) and L are necessary, which is already a widely accepted belief and practice in memetic algorithm (2) W or clustering can serve as selection, whose potential are obvious in Bag 4 when comparing the group with clustering/selection and the group without clustering/selection. However, the selection pressure via clustering should be harnessed more finely. In the paper [48], clustering based subset selection has been proved as effective inductive generational distance(IGD) based selection. (3) Comparison of two green lines in Bag 3 tells us that clustering or W helps speed up the convergence, namely, positional building blocks are effectively mined and used in that scenario.

| Bags | WXLs (g1) XLS (g2), no MWT | WXLs (g3) XLS (g4), +MWT | W S (g5) X S (g6), +MWT | LS (g7) LS (g8), +MWT | WXL (g9) XL (g0), +MWT |
|------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Bag 0 | ee              | e               | ee              | ee              | ee              |
| Bag 1 | e               | e               | ee              | e               | ee              |
| Bag 2 | ee              | ee              | ee              | ee              | ee              |
| Bag 3 | ee              | ee              | ee              | ee              | ee              |
| Bag 4 | ee              | ee              | ee              | ee              | ee              |

6.3. More about classical BB theory: combining biology inspired computing and mathematical tools of random walk and group theory

BB theory is a language/view to characterize GAs and MAs, like exploration and exploitation(R-IT) balance, when BB works well or R-IT balance performs well, GA and MA will do a good job. However, harness of BB is hard to achieve in its nature like R-IT balance, which may be released by both biological and mathematical sides.

Recently, a biological perspective on evolutionary computation[46] summarizes 6 directions, (1) openendness, (2) major transitions in organizational structure,(3) neutrality and random drift,(4) multi-objectivity,(5) complex genotype-phenotype mappings and (6) co-evolution. For (4) multi-objectivity, biology do the job implicitly, while computation works in an explicit way, which “departs from biology”[46]. The well designed SMO may help us to become more close to biology and towards “more natural high-level fitness functions and encouraging niche formation”[46].

Representation of permutations in PFSP serves a nice and straightforward computational model of group theory in algebra[47]. Any techniques of random walks[47] in group theory may help us to discover more specific local search as “nature” or KG2 in previous 6.2.

6.4. Two big unified pictures: decomposition styles, transfer family

**Picture 1.** decomposition styles.

Two well-known approaches of mathematical programming for combinatorial problems including PFSP are lagrangian relaxation(LR) and column generation(CG). When used to tackle combinatorial problems, like vehicle routing problem with time windows(VRP-TW) [36, 37], both two are deployed in decomposition styles. LR splits the original problem into independent subproblems, and CG turns the targeted problem into a master problem and a subproblem which again is decomposed into several independent problems.

Stepping into evolutionary or metaheuristics methods, MOEA/D and co-evolutionary algorithms[31] are wildly known for their decomposition power.

As to SMO or ETO_PFSP, it roughly divides the pipeline into 3 components, task 2, task 1.0 and task 1.1. For example, t2_wc = t2_wc + t_wc 1.0 + t_wc 1.1. Subproblems and decomposition also are developed here.

Then, **Picture 2.** transfer family(TF).

Both transfer learning (TL) and transfer optimization(TO) belong to the TF. The former includes 4 classes[50], TL for classification(TL1), TL for regression(TL2), TL for clustering(TL3) and TL for reinforcement learning(TL4). TO contains transfer bayesian optimization (also, it is TL2) and evolutionary based optimization of ETO(including our SMO or ETO_PFSP, MTO/MFO,…).

Actually, our ETO_PFSP includes a specific one of TL3 and a specific one of TO.

In TL, there exists[50] transductive TL, unsupervised TL and inductive TL(iTL). For inductive TL, the source and target domains are the same. And those two domains are also the same for our ETO_PFSP, that is, our framework shares common properties and settings with iTL, but things are more
complex in evolutionary scenarios, as domains are not enough, what matter more are the landscapes built by domains and operators (“parameter-transfer approach” in TL may relate to guided crossover probability within MTO setting). That is, we already see some deep connection between TL and TO. As fruitful results already achieved by TL may help the new born TO, attention to inspiration from TL to TO should not be abandoned.

7. Conclusion and future work

Considering major national needs within China’s context, our framework ETO_PFSP attempts to avoid scheduling production operations from scratch, not only contributing to “China’s industrial upgrading and transformation” [2] [16], but also heading towards the pledge of the China’s carbon neutrality.

Concerning with new frontier of ETO and evolutionary computation, our solution of ETO for SMO/Complex Optimization can reduce the complexity of multi-objective and even many objective problems, showing great potential scientific and academic values.

In the future, many directions are attractive and inspiring. First, extensive study of [K1,K2] may be helpful to the philosophy of ETO in Chapter 1. Then, extending to other problem pair in scheduling is also quite anticipated. At last, to disentangle the kinds of different knowledge learned / transferred is also important (maybe via fitness landscape tool of local optima network [39, 41]), which somehow like how the disentanglement of different features / representation[5][15] in deep learning/transfer learning[5,51] towards interpretability [17, 18, 19, 20, 21, 30, 40, 53, 54] of AI.

ACKNOWLEDGMENT

The authors thank Teachers Han Zhou (also, in our group of DAO, NEU) for their useful support and kindly help. Student Xu(one of the authors), also thanks a PhD student Zuocheng Li(also, in our group), for his useful discussion of building block theory. At last, student Xu expresses his thank to Prof. Yingping Chen(from National Yang Ming Chiao Tung University) for his insights on linkage learning genetic algorithms, which helps him(student Xu) understand the building block theory better. This research was supported by the Fundamental Research Funds for the Central Universities (N2128001).

REFERENCES

[1] Wu F, Yang C, Lan X, Ding J, Zheng N, Gui W, Gao W, Chai T, Qian F, Li D, Pan Y, Han J, Fu J, Liu K, Su S, Wu G (2018) Artificial intelligence: review and future opportunities. Bull of Nation Naturl Sci Foum of China[吴飞, 阳卓, 拇成光, 唐文员, 郑承南, & 桂卫华等. 人工智能的回顾与展望. 中国科学院学报.]
[2] Tang L, Meng Y (2021) Data analytics and optimization for smart industry. Frontiers of Eng Mana
[3] Tan KC, Feng L, Jiang M (2021) Evolutionary transfer optimization - a new frontier in evolutionary computation research. IEEE Comp Inte Magna
[4] Xu W, Zhang M (2018) Towards WARSHIP: combining brain-inspired computing of RSH for image super resolution. IEEE Clou Comp and Inte Syst
[5] Bengio Y (2011) Deep learning of representations for unsupervised and transfer learning. Workshop on Unsupervised & Transfer Learning
[6] Rodriguez A, Lao A (2014) Clustering by fast search and find of density peaks. Science

[7] Hank GR (1997) Learning gene linkage to efficiently solve problems of bounded difficulty using genetic algorithms. A PhD Thesis in The University of Michigan
[8] Chen Y (2006) Extending the scalability of linkage learning genetic algorithms. Part of the Studies in Fuzziness and Soft Computing book series
[9] Chang PC, Huang WH, Wu JL, Cheng TCE (2013) A block mining and recombination enhanced genetic algorithm for the permutation flow-shop scheduling problem. Inter J of Prod Econ
[10] Huang L, Feng L, Wang HD, Hou Y, Liu K, Chen C (2020) A preliminary study of improving evolutionary multi-objective optimization via knowledge transfer from single-objective problems. IEEE Inter Conf on Syst, Man, and Cybe
[11] Yuan Y, Ong YS, Gupta A, Tan PS, Xu H (2016) Evolutionary multitasking in permutation-based combinatorial optimization problems: realization with TSP, QAP, LOP, and ISP. IEEE Region 10 Conf
[12] Feng L, Ong YS, Lim M, Tsang IW (2015) Memetic search with interdomain learning: a realization between CVRP and CARP. IEEE Trans on Evol Comp, vol. 19, no. 5, pp. 644-658
[13] Wang X, Tang L (2017) A machine-learning based memetic algorithm for the multi-objective permutation flowshop scheduling problem. Comp & Oper Rese
[14] Li Y, Wang C, Gao L, Song Y, Li X (2020) An improved simulated annealing algorithm based on residual network for permutation flow shop scheduling. Comp & Inte Sys
[15] Shu J, Zhao L, Wang X, Zhao W, Huang M (2020) A novel deep Q-learning-based air-assisted vehicular caching scheme for safe autonomous driving. IEEE Trans Intelli Transport Sys
[16] Wang L, Wang J, Wu C (2018) Advances in green shop scheduling. Comp & Oper Rese
[17] Lei N, Luo Z, Yao ST, Gu DX (2018) Geometric understanding of deep learning. arXiv:1805.10451
[18] Bronstein M, Bruna J, Cohen T, Velivoki P (2021) Geometric deep learning: grids, groups, graphs, geodesics, and gauges.
[19] Zhu Y, Gao T, Fan L, Huang S, Edmonds M, Liu H, Gao F, Zhang C, Qi S, Wu Y, Tenenbaum J, Zhu S (2020) Dark, beyond deep: a paradigm shift to cognitive AI with humanlike common sense. arXiv:2004.00044
[20] E W (2017) A proposal on machine learning via dynamical systems. Comm. in Math & Stat
[21] Li D (2020) Ten questions for the new generation of artificial intelligence. CAAI Trans on Intell Sys (李德毅. 新一代人工智能十大课题. 2020)
[22] Littman ML, Aujuna I, Berger G, Boutilier C, Currie M, Velez FD, G. Hadfield, Horowitz IC, Isbell C, Kitano H, Levy K, Lyons T, Mitchell M, Shah J, Sloman S, Valor S, Walsh T (2021) Gathering strength, gathering storms: the one hundred year study on artificial intelligence (AI100) Study Panel Report. Stanford University, Stanford, CA, September, 2021. Doc: http://ai100.stanford.edu/2021-report
[23] Stone P, Brooks R, Brynjolfsson E, Calo R, Etzioni O, Hager G, Hirschberg J, Kalyanakrishnan S, Kamar E, Kraus S, Brown KL, Parkes D, Press W, Saxonian A, Shah J, Tambe M, Teller A (2016) Artificial Intelligence and Life in 2030, Doc: http://ai100.stanford.edu/2016-report
[24] Chen Y, Chuang C, Huang Y (2014) Inductive linkage identification on building blocks of different sizes and types. Inter Journ of System Scien
[25] Krasnogor N, Smith J (2005) A tutorial for competent memetic algorithms: model, taxonomy and design issues. IEEE Trans Evol Comput 9(5):474-488
[26] Yazdani D, Cheng R, Yazdani D, Branke J, Jin Y, Yao X (2021) A survey of evolutionary continuous dynamic optimization over two decades – Part B. IEEE Trans Evol Comput, pp(99):1-1
[27] Cotta C, Fernández AJ (2007) Memetic algorithms in planning, scheduling, and timetabling. Springer Berlin Heidelberg
[28] Deb K, Agrawal S (2002) Understanding interactions among genetic algorithm parameters.
[29] Ai-Sahaf H, Bi Y, Chen Q, Lensen A, Mei Y, Sun Y, Tran B, Xue B, Zhang M (2021) A survey on evolutionary machine learning. Journ of the Royal Soci of New Zeal, 2021, pp49:2, 205-228
[30] Yu Y, Chao Q, Zhou Z (2010) Towards analyzing recombination operators in evolutionary search. Interconfer on Parallel Prob Solv from Nature Comput

[31] Li K, Chen RZ, Fu G, Yao X (2017) Two-archive evolutionary algorithm for constrained multi-objective optimization. IEEE Trans Evol Comput, pp(99):165-181

[32] Wall MB (1996) A genetic algorithm for resource-constrained scheduling. PhD Dissertation, Massachusetts Institute of Technology, US

[33] Pinedo ML (2016) Scheduling: theory, algorithms, and systems. Springer

[34] Huang G, Huang GB, Song SJ, You KY (2015) Trends in extreme learning machines. Neural Netw

[35] Ruiz R, Pan QK, Naderi B (2019) Iterated greedy methods for the distributed permutation flowshop scheduling problem. Omega, 83(MAR.):213-222

[36] Tan KC, Lee LH, Zhu QL, Ou K (2001) Heuristic methods for vehicle routing problem with time windows. Artif Intell in Engin, 15(3):281-295

[37] Cordeau J, Desaulniers G, Desrosiers J, Solomon MM, Soumis F (2000) The VRP with time windows. Mono on Disc Math and Appli, SIAM, Philadelphia, Pa

[38] Watson J, Barbulescu L, Whitley LD, Howe AE (2002) Contrasting structured and random permutation flow-shop scheduling problems: search-space topology and algorithm performance. INFO Jour Compu

[39] Fieldsend J (2018) Computationally efficient local optima network construction. the Genetic and Evolutionary Computation Conference Companion

[40] Zhou Z, Yu Y (2006) A new approach to estimating the expected first hitting time of evolutionary algorithms. Proc of the 21st nation confer on Arti inte1, pp555-560

[41] Vérel S, Daolio F, Ochoa G, Tomassini M (2011) Local optima networks with escape edges. Intern Conf eron Arti Evol

[42] Bandaru S, Ng AHC, Deb K (2017) Data mining methods for knowledge discovery in multi-objective optimization. Expert Syst with Application, 70(mar.):139-159

[43] Liu XF, Xie YF, Li FB, Oui WH (2019) Admissible consensus for homogenous descriptor multiagent systems. IEEE Trans Sys, Man, and Cybernetics: Systems

[44] Gupta A, Ong YS, Feng L (2016) Multifactorial evolution: toward evolutionary multitasking. IEEE Trans on Evolu Computation

[45] Park J, Mei Y, Nguyen S, Chen G, Zhang M (2017) An investigation of ensemble combination schemes for genetic programming based hyper-heuristic approaches to dynamic job shop scheduling. Applied Soft Compu

[46] Miikkulainen R, Forrest S (2016) A biological perspective on evolutionary computation. Nature Machine Intelligence

[47] Pak I (1997) Random walks on groups: Strong uniform time approach. Harvard University

[48] Chen W, Ishibuchi H, Shang K (2021) Clustering-based subset selection in evolutionary multiobjective optimization. arxiv :2108.08453

[49] Xu W, Wang X, ETO meets scheduling: learning key knowledge from single-objective problems to multi-objective problem. Chinese Automation Congress, accepted, 2021.

[50] Pan SJ, Yang Q (2010) A survey on transfer learning. IEEE Trans on Know and Data Engin

[51] Pan Y, Li X, Yang Y (2017) A content-based neural reordering model for statistical machine translation. Commu. in Computer and Inform. Science

[52] Gui WH, Zeng ZH, Chen XF, Xie YF, Sun YB (2020) Knowledge-driven process industry smart manufacturing. Scien Sinica Inform (桂卫华、曾朝辉、陈晓方、谢芳芳、孙玉波, 知识驱动的流程工业智能制造). 中国科学: 信息科学

[53] Xu W, Wang X, Guo Q, Song X, Zhao R, Zhao G, Yang Y, Xu T, He D Towards KAB’S (Ⅲ): a proposal on ETO interpretability to disentangle the knowledge

[54] Xu W, Wang X, Guo Q, Song X, Zhao R, Zhao G, Yang Y, Xu T, He D Gathering strength, gathering storms: knowledge transfer via selection for VRPTW.