Independent Influence of Exploration and Exploitation for Metaheuristic-based Recommendations
Alexandre Bettinger, Armelle Brun, Anne Boyer

To cite this version:
Alexandre Bettinger, Armelle Brun, Anne Boyer. Independent Influence of Exploration and Exploitation for Metaheuristic-based Recommendations. The Genetic and Evolutionary Computation Conference (GECCO), Jul 2022, Boston, United States. 10.1145/3520304.3528972. hal-03655953

HAL Id: hal-03655953
https://inria.hal.science/hal-03655953
Submitted on 30 Apr 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Independent Influence of Exploration and Exploitation for Metaheuristic-based Recommendations

Alexandre Bettinger, Armelle Brun, Anne Boyer
UL CNRS LORIA
Nancy, Lorraine, FRANCE

ABSTRACT
Exploration and exploitation (E&E) of a search space are two fundamental processes in many fields of artificial intelligence. Indeed, when the search space is vast, it is important to ensure that many of its regions are examined, so as not to get trapped in a local optimum, but also that promising regions are examined more in depth in order to find good local optima. Influencing both processes is thus necessary. The literature has rarely proposed to approach the recommendation task as a vast search space problem. This paper introduces a metaheuristic-based recommendation approach with two contributions: an E&E influence process which is independent from the influenced algorithm and new indicators to represent and explain E&E. Performed on a genetic algorithm (GA) and on a reinforcement algorithm (RA), our experiments confirm that (1) the proposed influence process has a positive impact on evaluation criteria and on E&E which brings better recommendations, (2) proposed indicators contribute to represent E&E from new angles.

CCS CONCEPTS
• Computing methodologies → Heuristic function construction.

KEYWORDS
Recommendation, metaheuristic, evolutionary algorithm, exploration, exploitation, influence

ACM Reference Format:

1 INTRODUCTION
Recommending is the act of filtering information in order to target elements likely to interest one or more users. This task can be viewed as the processing of a vast search space which represents all possible recommendations. Depending on the recommendation context, a recommendation can take various forms such as single items, itemsets or item sequences. In what follows, we are interested in the itemset recommendation framework.

The search space can be crawled through an E&E process to find good recommendations [2]. Exploring means making sure that many regions of the search space are examined [3], so as not to get trapped in a local optimum. Exploiting means that if a promising region of the search space is found, it is examined more carefully [3] to see if it contains good local optima.

The literature highlights that the area of metaheuristics (MH) addresses the issue of proposing recommendations in huge search spaces [3]. Our work aims to contribute to E&E influence and explainability in the MH recommendation frame.

Studies have been conducted to influence the dynamics of E&E in order to ensure better search behaviors and thus find better recommendations regarding predefined evaluation criteria [3]. Influencing E&E implies to know when, why and how. Indeed, E&E have to be controlled regularly through a measure, and operations done to perform the influence must be justified. These elements contribute to explaining E&E behaviors.

The problem we aim to tackle in this work is: How to control, influence and explain E&E in the MH recommendation frame?

Our contribution is threefold. First, we propose an influence process independent of the influenced algorithm in order to promote the full expression of the influence and in order to not undergo structural and/or parametric modifications inside the influenced algorithm. Second, we propose a systematic influence of E&E, at each iteration of the influenced algorithm and as long as the itemsets do not comply with an E&E indicator. Finally, we introduce indicators to measure and represent E&E from new angles.

2 RELATED WORKS
In vast search spaces, filtering information is tedious and performing an exhaustive search is unceivable. Targeting elements to recommend requires to partially crawl the search space using E&E algorithmic mechanisms [2].

We propose a synthesis of three reviews [3][13][11] which address issues related to the E&E influence.

2.1 MHs for the Recommendation Task
Heuristic (H) methods refer to techniques for problem solving, learning and discovery where exhaustive research is not possible. MH methods are Hs integrated into iterative processes in order to increase their E&E capacities.

Our contribution falls within the domain of MHs because they are transparent and explainable, they perform well in hazardous environments and they are excellent for optimizing decisions.

MHs can be used to address the issue of proposing recommendations in huge search spaces. We found two interesting global surveys regarding MHs [4][5] and some specific contributions referring the use of MHs for recommendation.
Many works have been interested in the trade-off between E&E, without leading to a complete or general answer. The recommendation frame is highly concerned by these problems. Contributions aiming to explain, measure, control and influence E&E are expected.

3 CONTRIBUTIONS

We propose a process to control and influence E&E of a search algorithm (called influenced algorithm). From our view, the influence processes proposed in the literature do not guarantee a highly impactful expression of the influence. One contribution of this work thus aims to give more power to the influence process by designing an independent process from the influenced algorithm. This independence makes that this influence process can be used on most of MHs, without major adaptation.

We choose to present an evolutionary algorithms (EA) among all available MHs because GA are popular. EA generally exploits arrays of digits, traditionally representing recommendable itemsets. Digits can take the value 0 (non-recommendation of the item) or 1 (recommendation of the item). The algorithm iteratively performs variations: generates, crosses and transforms arrays, while trying to optimize and keep the best of them with regard to predefined constraints and evaluation criteria.

In addition, we propose new indicators with the objective to have an understandable representation of E&E behavior.

3.1 E&E Influence Process

In the literature, the influence process has characteristics that limit its impact on E&E. It (1) strongly depends on the influenced algorithm, like in the parameter optimization case (Algorithm 1), (2) does not influence enough, for example when it does not have its own variation operators, (3) does not guarantee any form of compliance with the E&E indicator at each iteration of the influenced algorithm. In this work, we propose an influence process that (1) is dissociated from the influenced algorithm, (2) has its own variation operators, (3) seeks to guarantee the compliance of the solutions with the E&E indicator at each iteration of the influenced algorithm. Thus, only the itemsets are possibly modified and returned to the influenced algorithm, the latter showing no structural and/or parametric modifications. It may bring different decisions regarding E&E influence, which deserves to be studied.

The indicator is the measure of E&E used to decide whether to force exploitation, exploration or both with additional variations. An example of E&E indicator represents the diversity of the solutions found so far by the influenced algorithm. The approach we propose is designed to use any indicator.

The operating principle of the influence process is based on an independent function called during the run of the influenced algorithm (Algorithm 2). In the literature the influence is operated only once per iteration but, in our view, right after an influence decision, the elements used for taking this decision could make it possible to decide to influence again. We thus propose an E&E control, carried out after each influence variation process which is operated as long as the itemsets are not compliant with the limits.
of the E&E indicator. The influence process thus guarantees the
itemsets compliance with the E&E indicator within each iteration.
We consider that doing so allows to dose out the influence impact,
the possible issue being to perform a too impacting influence that
disfigures what the influenced algorithm does regarding E&E. We
must also note that the influence process will inevitably increase
the computational cost as it brings more treatments.

Concretely, all solutions of the influenced algorithm are passed
to the influence function (Line 5), which will transform them iterat-
tively according to the evolution of the E&E indicator. When the
indicator position itself in a specific range of values, the influence
function stops the transformations and returns all the solutions to
the influenced algorithm, which continues its execution until the
next call of the influence function. Note that the influence vari-
tions are made as long as the indicator is outside the range. It is
therefore necessary to replace the values of the limits in line with
the evolution of the E&E indicator and thus to converge towards
stabilization around the observed values of the E&E indicator. To
do this, after each iteration of the influence function, the minimum
and maximum limits of the E&E indicator are adjusted using a
parameter added or removed from the average value of the E&E
indicator observed so far.

3.2 Indicators to Influence and Explain E&E
As suggested in the literature, E&E would deserve to be represented
from new angles [3][13][11]. In response, we propose three types
of indicators that measure and represent E&E behaviors and that
can be used within the influence function previously introduced.

3.2.1 Constraints Completion (CC). The CC indicator assesses
the constraints completion within itemsets by treating constraints as
volumes to be filled. To evaluate this indicator, for each itemset the
difference between its constraint score (cs) and constraint maximum
limit (cml) is evaluated, as well as the average of all differences
obtained (Equation 1). The larger this gap, the more space to fill
among all itemsets as it is potentially possible to add more items in
some of them, while meeting constraints. Indeed, the E&E process
did not allow for inserting these items so far.

\[ CC = \frac{cml - cs}{2} \] (1)

3.2.2 E&E Power. The EEP indicator represents the E&E power as
the amount of exploration towards exploitation. We consider that
each time a digit is passed to 1 for the first time, this constitutes an
exploration. An iteration discovering at least one new digit is an
exploration iteration and a vector containing at least one new digit
is an exploration vector.

We evaluate the number of exploration iterations divided by the
number of exploitation iterations (%eit), the number of exploration
vectors divided by the number of exploitation vectors (%evec) and
the number of digits newly discovered divided by the number of
digits already discovered (%ndd). EEP is the average of these three
elements (see Equation 2).

\[ EEP = \frac{\%eit, \%evec, \%ndd}{3} \] (2)

Eventually, a register of undiscovered digits can be used to for-
mally define an E&E phase. Let p be an E&E phase that can take the
values [exploration, exploitation]. Let ndd be the register of digits
not yet discovered. This register is updated through iterations by
removing digits from it once they are used (passed to value 1) for
the first time inside vectors. Let ie be an end of iteration marker of
the influenced algorithm. ndd(ie) therefore corresponds to ndd at
the end of iteration marker ie. So we can define p as in Equation 3.

\[ p(ie + 1) = \begin{cases} \text{exploration,} & \text{if } ndd(ie) > ndd(ie + 1) \\ \text{exploitation,} & \text{if } ndd(ie) = ndd(ie + 1) \end{cases} \] (3)

3.2.3 Temporality. We propose to represent the alternation of E&E
iterations as gaps between time values. The temporal differences
(gaps) between starting time of two iterations are evaluated and
bring out the minimum (ming) and maximum (maxg) gaps, the
difference between the minimum and the maximum gap (DMMG)
(Equation 4), the average gap (AG) (Equation 5). DMMG allows to
evaluate most extreme time gaps between exploration iterations. AG
implicitly represents the quantity of exploration iterations which
implicitly represents their regularity. Depending on the context, a
high or low regularity is expected.

\[ DMMG = \max g - \min g \] (4)
\[ AG = \frac{\text{gaps}}{\text{iterations}} \] (5)

4 EXPERIMENTS

4.1 Dataset and Evaluation Measures
Our experiments are carried out on the ‘Movielens 25M’ [6] dataset.

Influence functions are applied on a traditional GA that serves
as a baseline for comparisons. The objective is to show how it
performs when influenced using different E&E indicators.

4.2 Evaluation Criteria
We decide to evaluate our influence process by considering several
criteria related to performance, quality and time, that are either
those introduced section 3.2 or proposed in the literature [8][7].

4.2.1 Performance Criteria. An item score is defined as the weighted
sum of its normalized attribute values and an itemset score is de-

dined as the sum of the scores of its items. Convergence Iterations
(CI): Average iterations required for reaching convergence points.
Max Score Iterations (MSI): Average iterations required to reach
maximum scores. Convergence Score (CS): Average maximum score
reached on convergence points. Max Score (MS): Average maximum
score reached. Execution Time (ET): Average execution time.

4.2.2 Quality Criteria. All Vectors Coverage (AVC): Coverage con-
idering all generated vectors. It represents the percentage of digits
that are at least once at 1 in the vectors considered. The higher
the coverage, the better it means that the search process has found many items. Constraints Completion (CC): Average
constraints completion percentage. It represents how well recom-
ended items fill the constraints. E&E Power (EEP): Part of explo-
ration towards exploitation. See section 3 for more details.
4.2.3 Temporal Criteria. Min-Max Gap (DMMG): Difference between the minimum and the maximum temporal gap between two exploration iterations. Mean Gap (AG): Average temporal gap between two exploration iterations.

4.3 Experiments

Aggregated percentages are shown in Table 1. The $\% \uparrow$, $\% \rightarrow$ and $\% \downarrow$ lines refer respectively to the percentages of increased, unchanged and decreased evaluation criteria results towards baseline algorithm, considering all execution cases. To complement this information, we perform ranksum tests represented by the $\%$Pass line showing the percentage of passed ranksum tests considering all execution cases. Let us precise that the null hypothesis of a ranksum test is that evaluation criteria values, of the non influenced algorithm and of one of its influenced version, come from the same distribution. If the null hypothesis is rejected, it is a Pass ($p$-value $< 0.05$) and otherwise it is a Fail ($p$-value $\geq 0.05$).

<table>
<thead>
<tr>
<th>Performance</th>
<th>Quality</th>
<th>Temporal</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI</td>
<td>MSI</td>
<td>CS</td>
</tr>
<tr>
<td>$% \uparrow$</td>
<td>62</td>
<td>65</td>
</tr>
<tr>
<td>$% \rightarrow$</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>$% \downarrow$</td>
<td>37</td>
<td>35</td>
</tr>
<tr>
<td>$%$Pass</td>
<td>17</td>
<td>20</td>
</tr>
</tbody>
</table>

Significant ratio values for most of the criteria show a positive impact of the influence process. Regarding performance criteria, the influence process makes the algorithm reach better scores at the expense of the number of iterations and execution time. Regarding quality criteria, the influence process makes the baseline algorithm evaluate more solutions. As users preferences are encoded in constraints and as CC remains in almost all cases at least equal to the baseline, we can conclude that the proposed influence process allows to find better recommendations. Besides, the influence process has a significant impact on EEP by favouring exploitation and a noticeable impact on E&E regularity.

To summarize, the influence process has a positive impact on the search behavior. We note that a detailed results analysis shows that some proposed influence cases perform better than the others. We also note that the proposed influence does not have a positive impact in all contexts, this was expected as it depends on the search space structure and on set of constraints. We also stress that results confirm that the influence process does not induce random behaviors as scores remain at least equals to the baseline.

Moreover, we confirm that proposed influence indicators provide explanations regarding E&E. Indeed, they represent E&E behaviors from new angles as shown with CC, EEP, DMMG and AG.

To show the adaptability of the influence process, we apply it to a RA. Results show a positive impact of the influence. Similar conclusions to the GA case can be drawn.

5 DISCUSSION AND FUTURE WORK

We highlight here the genericity of the proposed influence process.

Firstly, the independent influence proposed is adaptable to genetic programming, to MHs and more generally to any algorithm based on the E&E of the search space.

Besides, the moment where the call of the influence function occurs impacts E&E and depends on the influenced algorithm as well as on the way the influence function is designed. We have chosen to position it before the end of each iteration of the influenced algorithm, this choice can be debated.

Moreover, influence variations can be performed in many ways, depending on the implementation context.

To continue, the proposed influence process could include too much randomness and thus alter the influenced algorithm enrichment. Our results show that if used properly, this is not an issue.

Finally, our contributions could be analyzed against other evaluation criteria.

Our future work will be marked by developments around new E&E influence processes and indicators. We will also perform experiments using other MHs to support that conclusions do not only stand for genetic programming and reinforcement. In addition, we intend to carry out more in-depth and diverse experiments in the recommendation frame to thorough study the characteristics of the recommendations provided. Finally, we will keep up with our work regarding the explainability of E&E behaviors.

ACKNOWLEDGMENTS

This work is carried out within the framework of the PEACE agreement with the French Rectorate of the Nancy-Metz Academy as well as with the French Ministry of National Education and Youth.

REFERENCES


