# A Comparison of Operator Utility Measures for On-line Operator Selection in Local Search

Nadarajen Veerapen[1], Jorge Maturana[2], and Frédéric Saubion[1]

[1] LUNAM Université, Université d'Angers, LERIA, Angers, France
firstname.lastname@univ-angers.fr
[2] Instituto de Informática, Universidad Austral de Chile, Valdivia, Chile
jorge.maturana@inf.uach.cl

**Abstract.** This paper investigates the adaptive selection of operators in the context of Local Search. The utility of each operator is computed from the solution quality and distance of the candidate solution from the search trajectory. A number of utility measures based on the Pareto dominance relationship and the relative distances between the operators are proposed and evaluated on QAP instances using an implied or static target balance between exploitation and exploration. A refined algorithm with an adaptive target balance is then examined.

## 1 Introduction

An increasing number of solving techniques have been proposed to address larger and more complex optimization problems but they are often difficult to adapt and to tune for a given problem. In fact, efficient solving tools have become out of reach for practitioners. Among the possible solving techniques, metaheuristics are now widely to efficiently solve optimization problems. Nevertheless, attempting to design increasingly efficient metaheuristics often results in highly complex systems which require a non-negligible amount of expert knowledge to use, for instance to wisely choose the method's required parameters.

A relatively recent avenue of research is the design of generic high level control strategies in an attempt to make optimization techniques more user-friendly [3]. A classification of these different approaches can be found in [5]. In general only one criterion, solution quality, is considered. Concerning the control of parameters, the most advanced techniques were first developed in the context of evolutionary computation [6]. A number of operator selection strategies for genetic algorithms (adaptive operator selection) are presented in [4]. In [7], operator selection techniques were proposed to handle simultaneously two criteria in the evaluation of the operators: quality and diversity of the population.

We focus on local search algorithms to solve combinatorial optimization problems. In a previous work [10], we have proposed a general framework to control dynamically the search process of a local search algorithm targeted at problems that can be modeled as permutations. In this paper, we improve this mechanism by introducing new performance evaluation techniques and more sophisticated and dynamic control features.

The paper has five more sections. Section 2 looks at operator control in local search. Section 3 describes the different utility values used in operator selection. Section 4 deals with the experiments. Section 5 looks at an attempt at adaptively changing the utility value parameter. Finally the last section is the conclusion.

## 2 Operator Control for Local Search

The defining feature of a good local search algorithm is the efficient exploration of the search space in order to find the optimal solution. This requires striking a balance between two generally conflicting objectives: exploitation (converging towards a local optimum) with exploration (suitably sampling different areas of the search space). One way of achieving this balance is by controlling the basic operations (moves or operators) which drive the solution around the search space.

Our aim is to select an appropriate operator out of a set of operators to be applied at each iteration of the local search algorithm. In order to determine the likeliness of an operator to be useful, in terms of both exploitation and exploration, its previous behavior needs to be recorded and analyzed. Figure 1 shows how the operator control interacts with the local search algorithm.

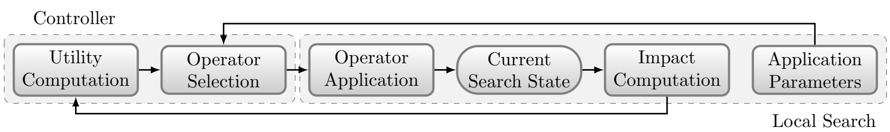


**Fig. 1.** Overview : Operator control (left) and local search algorithm (right).

*Impact Computation.* Quality is measured directly as the change in the objective function. Measuring diversity is relatively straightforward in evolutionary algorithms but less clear in single point algorithms. Here we consider a sliding window containing the last solutions in the search path and measure the difference between them and the current candidate solution $c = op(s)$. This difference is computed at the variable-value couple level: the less frequent the occurrences of the candidate solution's variable-value couples in the path, the greater the distance between them. The following equation formalizes this notion. Let $P_{i,j}$ be the path from iteration $i$ through $j$, $i \leq j$. Then $d^P(c, P_{i,j}) = 1/n \times \sum_{k_c=1}^{n} (1 - occ(P_{i,j}, (k_c, \pi_{k_c}))/|P_{i,j}|)$ where $occ(P_{i,j}, (a, b))$ returns the number of times the variable-value couple $(a, b)$ is found in $P_{i,j}$.

## 3 Operator Selection and Utility

An operator is selected with a probability proportional to some utility value which is meant to be a reflection of its previous performance. Each operator has its own fixed-length sliding window which keeps track of the quality difference and distance value for its $m$ last applications. The sliding windows are initialized by one application of each operator at the beginning of the search. The utility of an operator is based on the average of the quality and distance values.

Given two vectors $u$ and $v$ of equal cardinality $p$ and considering a maximization problem, $u$ *dominates* $v$ if $u_k \geq v_k, \forall k \in \{1, \ldots, p\}$ with at least one strict inequality. This is often referred to as Pareto dominance and denoted by $u \succ v$.

The Pareto-dominance-based utility $U_P$ of operator $o$ among the set $O$ of $n$ operators would then be defined as $U_P(o) = |\{o' | o' \in O, o \succ o'\}| + \epsilon$ where $\epsilon$ ensures a non-zero utility value. This utility assignment scheme (used in [10]) does not allow for a commanded balance between exploitation and exploration.

We now propose a number of ways of introducing a weight, $\alpha$, in order to influence the balance. Given two operators $o_1$ and $o_2$ defined by the quality-distance couples $(q_1, d_1)$ and $(q_2, d_2)$, we define the weighted rectilinear displacement from $o_1$ to $o_2$ as $d_\alpha(o_1, o_2) = \alpha(d_1 - d_2) + (1 - \alpha)(q_1 - q_2)$

Based on this metric, we define three utilities: the sum of displacements to all other operators $U_\alpha^{\Sigma}(o) = max(0, \sum_{i=1}^{n} d_\alpha(o, o_i))$, the sum of positive displacements to all other operators $U_\alpha^{\Sigma+}(o) = \sum_{i=1}^{n} max(0, d_\alpha(o, o_i))$, the sum of displacements to all other dominated operators $U_\alpha^{\Sigma\succ}(o) = \sum_{i=1, o \succ o_i}^{n} d_\alpha(o, o_i)$. We also use the very simple weighted sum of operator quality and distance $U_\alpha(o) = \alpha d + (1 - \alpha)q$.

Displacements, and their sum, are useful in the sense that they are a means of describing quantitatively (the magnitude) and, to a lesser extent, qualitatively (the sign or direction) the relationship between each operator and the rest. In addition, the weight naturally introduces a quantifiable bias towards either exploration or exploitation.

## 4 Experiments for Weighted Operator Utility

The different utility values described in the previous section are tested on a very classic permutation problem: the Quadratic Assignment Problem (QAP) which models the problem of finding a minimum cost allocation of $N$ facilities into $N$ locations, taking the costs as the sum of all possible distance-flow products.

*Experimental Settings.* Each operator is a combination of a neighborhood and a selection function. We use a single basic neighborhood which swaps the values of two variables. The selection functions used are random selection, first improving neighbor, best neighbor, random selection among the 5 best neighbors, and best among $k$ neighbors.

A population of twelve operators is defined. Ten of these do not change the solution configuration much (at most 6 variables are affected): half are intensification oriented, half are exploration oriented. The last two are extremely perturbative operators which randomly swap 25% and 50% of the variables.

For each instance, each algorithm is run thirty times starting with the same thirty different random solutions and a maximum of 40 000 iterations are allowed per run. All sliding windows have an arbitrary fixed length of 100.

*Analysis of Results.* Table 1 reports the results for all the experiments on the QAP (instances from QAPLIB [2]) in this paper. In this section all columns except the next-to-last one is of interest to us. They report the instances, their best known value and the results for the different utility values with fixed weights: $U_\alpha$ with $\epsilon = 0.1$ (the values 0.001, 0.05, 0.2 and 1 were also tested but 0.1 proved the best across almost all instances) and $U_\alpha^{\Sigma}$, $U_\alpha^{\Sigma+}$, $U_\alpha^{\Sigma\succ}$ and $U_\alpha$ with $\alpha$ taking values 0.2, 0.5 or 0.8. The last column gives, as a comparison, the results obtained with Robust Tabu (RoTS) [9], a dedicated local search algorithm for the QAP. The values for each algorithm express the average percentage difference above the best known values. Bold font indicates the best values and italics indicate results which are within 0.05 % of the best (RoTS results are not considered because they are always better or equal).

From Table 1 it is clear that $U_\alpha^{\Sigma+}$ is the best among the displacement-based utility values tested and it provides consistently good results over various $\alpha$. Pareto-dominance-based selection, $U_P$, also provides good results with respect to all other selection methods. Our previous work had shown that it worked slightly better than uniform selection if the operators were relatively "good" and outperformed it when a single "very good" operator was added. Here we can see that it remains useful despite the presence of highly disruptive operators.

When considering $U_P$ and $U_\alpha^{\Sigma+}$ selections side-by-side, the latter is generally equivalent to the former for lower $\alpha$ values and only really better on tai$xx$a instances. Following this observation, and also because different classes of instances seem to require different values of $\alpha$ to obtain the best results, a method for adapting $\alpha$ and obtaining better results is described in the next section.

If we consider $U_\alpha^{\Sigma}$, the problem is that accepting negative displacements often negates the other positive displacements thus producing results that are, usually, worse than uniform selection. The poor performance of $U_\alpha^{\Sigma\succ}$ may be explained by the fact that since it is based on Pareto dominance and there is no $\epsilon$ to ensure a minimum selection probability, the operators at both ends of the exploration-exploitation spectrum have no real chance of being selected because they usually do not dominate any other operator. Finally, it appears that $U_\alpha$ might be too simple and shows that the relationship between operators can be useful when used appropriately (in $U_P$ and $U_\alpha^{\Sigma+}$).

The next section looks at how the weight $\alpha$ can be varied during the search.

## 5 Adaptive Parameter Values for Operator Control

We consider the "correct" diversity (CD) strategy [8]. The CD strategy uses the value of the quality of the solutions in the population as a means to assess the diversity of population. If the number of solutions having the same quality is above a certain threshold $T_{max}$ then it is assumed that the population is too homogeneous and the commanded diversity is incremented by some step $s_{inc}$. Symmetrically, if the number of solutions having the same quality is below another threshold $T_{min}$ the commanded diversity is decremented by some step $s_{dec}$ (solutions are haphazardly distributed; exploitation is not strong enough).

*Experiments.* To tune the CD parameters we use F-Race [1] an off-line tuning algorithm. We first tested 320 parameter combinations. The winning parameters ($T_{max} = 0.3$, $T_{min} = 0.25$, $s_{inc} = 0.0001$, $s_{dec} = 0.1$) were at either end of the available domain for each parameter. One could thus assume that a better combination of parameters might be obtained by extending their domain. We therefore ran a second race with new parameter domains (144 combinations) and obtained a new winner ($T_{max} = 0.35$, $T_{min} = 0.15$, $s_{inc} = 0.0001$, $s_{dec} = 0.01$) which was relatively different from the previous one and did not benefit from the new values at the extremities of each domain. For both winners the distributions of results were statistically equivalent. This leads us to believe that the strategy parameters need only be within some tight domain (and not one specific value) to obtain the best results.

*Results and Discussion.* The results are presented in Table 1 under the $U_\alpha^{\Sigma+}CD$ column. It seems clear that $CD$ is better than the other selection methods, or within 0.05 % of the best, on most instances in terms of raw results. This superiority is further confirmed by a Wilcoxon signed-rank test with 95 % confidence level. If we compare $U_\alpha^{\Sigma+}CD$ with the best values across the different $\alpha$ for $U_\alpha^{\Sigma+}$ both distributions are statistically equivalent. This leads us to conclude that the $CD$ strategy is good enough to produce results equivalent to the best results of $U_\alpha^{\Sigma+}$ with fixed $\alpha$ values.

**Table 1.** Results for QAP instances.

| Instance | BKV | Uniform | $U_P$ | $U_\alpha^{\Sigma+}$ | $U_\alpha^{\Sigma}$ | $U_\alpha^{\Sigma\succ}$ | $U_\alpha$ | $U_{\alpha CD}^{\Sigma+}$ | RoTS |
|---|---|---|---|---|---|---|---|---|---|
| chr25a | 3796 | 20.18 | **10.67** | 13.78<br>14.94<br>12.11 | 33.53<br>30.75<br>28.70 | 34.05<br>28.68<br>29.75 | 31.34<br>30.18<br>16.44 | 12.45 | 7.09 |
| kra30a | 88900 | 2.49 | 0.79 | 1.57<br>1.63<br>0.89 | 5.14<br>4.39<br>3.75 | 4.67<br>4.55<br>4.42 | 5.24<br>4.97<br>2.20 | **0.61** | 0.06 |
| kra30b | 91420 | 1.11 | 0.21 | *0.16*<br>0.45<br>0.32 | 3.06<br>2.78<br>2.50 | 3.32<br>3.03<br>2.37 | 3.25<br>2.63<br>0.68 | **0.13** | 0.02 |
| nug20 | 2570 | 0.12 | *0.01* | **0.00**<br>*0.03*<br>**0.00** | 1.02<br>0.56<br>1.15 | 1.09<br>0.89<br>1.45 | 1.74<br>0.65<br>0.07 | *0.01* | 0.00 |
| nug30 | 6124 | 1.24 | 0.20 | 0.31<br>0.19<br>0.39 | 1.67<br>1.43<br>1.60 | 1.27<br>1.54<br>1.71 | 1.86<br>1.75<br>0.68 | **0.11** | 0.01 |
| sko42 | 15812 | 2.28 | 0.29 | *0.19*<br>0.28<br>0.67 | 1.91<br>1.38<br>2.01 | 1.65<br>1.63<br>2.14 | 1.93<br>1.59<br>1.48 | **0.16** | 0.03 |
| sko49 | 23386 | 2.48 | 0.36 | **0.21**<br>0.27<br>0.81 | 1.37<br>1.34<br>2.31 | 1.46<br>1.60<br>1.74 | 1.57<br>1.42<br>1.91 | *0.24* | 0.13 |
| tai30a | 1818146 | 2.59 | 1.26 | 1.17<br>1.27<br>1.68 | 2.05<br>1.86<br>3.18 | 2.16<br>1.78<br>3.41 | 3.31<br>1.87<br>2.33 | **0.91** | 0.51 |
| tai50a | 4941410 | 4.20 | 2.16 | **1.58**<br>*1.59*<br>2.83 | 2.13<br>2.61<br>4.11 | 2.27<br>2.80<br>4.11 | 3.40<br>2.34<br>3.82 | 1.66 | 1.39 |
| tai30b | 637117113 | 0.43 | **0.13** | 0.44<br>0.35<br>*0.16* | 6.65<br>3.90<br>3.49 | 5.21<br>3.53<br>1.74 | 5.27<br>4.83<br>0.34 | *0.15* | 0.03 |
| tai50b | 458821517 | 2.36 | 0.25 | 0.30<br>0.39<br>0.37 | 4.14<br>3.13<br>2.56 | 4.33<br>3.92<br>2.66 | 5.42<br>4.33<br>1.60 | **0.18** | 0.15 |

## 6 Conclusion

In this paper we have presented different alternatives for the selection of operators in Local Search. The main contribution of the paper was the investigation of weighted utilities which allow a target balance to be set between exploration and exploitation. Using static weights the best of them was competitive when compared to the previously proposed Pareto-dominance-based utility. An adaptive strategy for setting the weight was investigated and proved to provide improved results.

In future works we wish to look at more advanced on-line parameter setting strategies. Another avenue of research is testing the existing proposed methods with academic problems such as the One-MAX and long-path problems, whose properties are well understood, in order to have a better theoretical understanding of the methods.

*Acknowledgments.* This work was supported by Microsoft Research through its PhD Scholarship Programme and by CONICYT–ECOS-Sud (Project C10E07).

## References

1. Birattari, M.: The Problem of Tuning Metaheuristics as seen from a machine learning perspective. Ph.D. thesis, Université Libre de Bruxelles, Brussels, Belgium (Dec 2004)
2. Burkard, R.E., Karisch, S.E., Rendl, F.: QAPLIB – a quadratic assignment problem library. Journal of Global Optimization 10, 391–403 (1997)
3. Burke, E., Kendall, G., Newall, J., Hart, E., Ross, P., Schulenburg, S.: Hyper-heuristics: An emerging direction in modern search technology. In: Glover, F., Kochenberger, G. (eds.) Handbook of Metaheuristics, International Series in Operations Research & Management Science, vol. 57, pp. 457–474. Springer New York (2003)
4. Fialho, Á.: Adaptive Operator Selection for Optimization. Ph.D. thesis, Université Paris-Sud 11, Orsay, France (Dec 2010)
5. Hamadi, Y., Monfroy, E., Saubion, F.: What is autonomous search? In: van Hentenryck, P., Milano, M. (eds.) Hybrid Optimization, Springer Optimization and Its Applications, vol. 45, pp. 357–391. Springer New York (2011)
6. Lobo, F.G., Lima, C.F., Michalewicz, Z. (eds.): Parameter Setting in Evolutionary Algorithms, Studies in Computational Intelligence, vol. 54. Springer (2007)
7. Maturana, J., Lardeux, F., Saubion, F.: Autonomous operator management for evolutionary algorithms. J. Heuristics 16(6), 881–909 (2010)
8. Maturana, J., Saubion, F.: On the design of adaptive control strategies for evolutionary algorithms. In: Monmarché, N., Talbi, E.G., Collet, P., Schoenauer, M., Lutton, E. (eds.) Artificial Evolution. LNCS, vol. 4926, pp. 303–315. Springer Berlin / Heidelberg (2008)
9. Taillard, É.D.: Robust taboo search for the quadratic assignment problem. Parallel Computing 17(4-5), 443–455 (1991)
10. Veerapen, N., Saubion, F.: Pareto autonomous local search. In: Coello, C. (ed.) Learning and Intelligent Optimization. LNCS, vol. 6683, pp. 392–406. Springer Berlin / Heidelberg (2011)