On the Effectiveness of Restarting Local Search

Aldeida Aleti, Mark Wallace
Faculty of Information Technology, Monash University, Australia

Markus Wagner
School of Computer Science, The University of Adelaide, Australia

ABSTRACT

Premature convergence can be detrimental to the performance of search methods, which is why many search algorithms include restart strategies to deal with it. While it is common to perturb the incumbent solution with diversification steps of various sizes with the hope that the search method will find a new basin of attraction leading to a better local optimum, it is usually unclear whether this strategy is effective. To establish a connection between restart effectiveness and properties of a problem, we introduce a new property of fitness landscapes termed *Neighbours with Similar Fitness*. We conjecture that this property is true for many PLS-complete problems, and we argue that the effectiveness of a restart strategy depends on this property.

ACM Reference Format:
Aldeida Aleti, Mark Wallace and Markus Wagner. 2021. On the Effectiveness of Restarting Local Search. In *Proceedings of the Genetic and Evolutionary Computation Conference 2021 (GECCO '21).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

A wide variety of techniques has been developed for tackling large scale combinatorial optimisation problems. Search methods, such as Evolutionary Algorithms [1], Ant Colony Optimisation [7] and Local Search [11] are typically used to achieve the required scalability in challenging problems for which it is hard to find optimal, or even just "good enough" solutions. The majority of these methods involve steps where the state of the algorithm is modified in some way to escape a local optimum, also known as a restart strategy. This is typically used to avoid premature convergence, which is when the search method converges (usually very early in the search) to a local optimum of poor quality [10].

To study how the structure of fitness landscapes impacts the effectiveness of search strategies, much of the existing research focuses on fitness landscape characterisation metrics that measure properties of the fitness landscape and relate them to the effectiveness of search algorithms. Existing metrics measure landscape properties such as the number of local optima, the size of the basin of attraction, and the size of the plateaus. While there have been many works showing how these properties correlate with the effectiveness of a wide range of solvers on a wide range of problems (see, e.g. [9]) there are only few fundamental studies tying landscape features to restart strategies. For example, for simple but nevertheless fundamentally important problems like OneMax and Jump, it was recently shown that drift arguments can be used to restart algorithms in combination with population size control [2]. Similarly, for a function comprising a plateau and a slope, it was shown that a particular scheme (that initially uses a number of parallel runs) needs to take into account characteristics of the function [5]. For more complex problems, however, there do not seem to be any studies yet that establish a connection of fundamental landscape features to restart performance.

Most search algorithms implement restart during the search, as a mechanism for resetting the search and preventing premature convergence [6]. However, there is little understanding on when these methods work. We introduce a new property of fitness landscapes, *Neighbours with Similar Fitness* (NSF), which means that neighbours in the search space tend to have similar fitness compared to non-neighbours. A neighbour is a solution that can be reached in one step by the search method from its current position, and a NSF neighbourhood is one that tends to link solutions with similar fitness. We provide a formal definition of NSF, and argue that NSF generally holds for many combinatorial optimisation problems and the search operators designed for algorithms that tackle them, hence a random restart strategy would be the most suitable.

2 NEIGHBOURS WITH SIMILAR FITNESS

Neighbours with Similar Fitness (NSF) is a property of fitness landscapes that states that two neighbouring solutions have a higher probability of having similar fitness values than two non-neighbouring solutions. While many search methods rely on gradients in the landscape, NSF is defined for discrete fitness functions where gradients do not exist. However, the intuition is similar: NSF holds if neighbours have similar fitness, which is similar to requiring a fitness function on the real numbers to be continuous. The degree to which a landscape has NSF is related to the gradient. However, there is an added dependency on the distribution of solutions in the search space which is normally absent in the discussion of landscapes.

We represent a combinatorial optimisation problem (S, f) as a finite search space S with a finite range of fitness values $\{f(s) : s \in S\} \subseteq \{v_{\min} \dots v_{\max}\}$. Without loss of generality, we consider maximisation problems in the following, i.e., solutions with higher fitness values are considered to be better.

We define $|f_v|$ to be the number of solutions in the search space with fitness value v. The search space size is equal to

GECCO '21, July 10–14, 2021, Lille, France

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

$|S| = \sum_{v \in V} |f_v|$. Given a solution with fitness v we would like to know the number of solutions with fitness difference by δ, defined as $f_{v,\delta}| = |\{s \in S : f(s) = v \pm \delta\}|$, where $\delta \in \Delta$, and Δ is the set of all differences among fitness values in V.

This tells us, for a solution with a particular fitness, how different the other solutions are. This also allows us to calculate the proportion of solutions that differ in fitness by δ from a given fitness v as $p_{v,\delta} = (|f_{v,\delta}|)/(|S|)$.

A fitness landscape associated with the problem (S, f) is defined as (S, f, N), where N is the neighbourhood function which identifies the solutions that can be reached in one search step from each other solution. We refer to the set of neighbours N_v of solutions with fitness v: $N_v = \{s' \in N(s) : s \in S \cap f(s) = v\}$. Note that this is the union of all the neighbourhoods of solutions with fitness v. We define the number of neighbours of solutions with fitness v, with fitness greater or lower by δ to be: $|N_{v,\delta}| = |\{s \in N : f(s) = v \pm \delta\}|$ The proportion of neighbours (solutions in N_v) that differ from v by δ, is then $pn_{v,\delta} = \frac{|N_{v,\delta}|}{|N_v|}$.

We use the expressions $p_{v,\delta}$ and $pn_{v,\delta}$ in the following definition of the Neighbours with Similar Fitness (NSF) property. The property states that neighbours tend to have more similar fitness than non-neighbours.

Definition 1. *A landscape has the* Neighbours with Similar Fitness *(NSF) property if for all fitness values v the value of the expression $pn_{v,\delta}/p_{v,\delta}$ decreases monotonically with increasing $\delta \in \Delta$.*

This property means that for each fitness value v, the probability that a neighbour of s has fitness $f(s)$ close to v is higher than the probability of a solution in the search space as a whole having fitness close to v. As the fitness value difference $\delta \in \Delta$ grows, the difference between the probability $pn_{v,\delta}$ of a neighbour with fitness differing by δ from v, and the probability $p_{v,\delta}$ of a solution in the whole search space with fitness differing by δ from v, monotonically decreases.

In problems that exhibit the NSF property, larger perturbations are required for an effective restart. In NSF, solutions of the same fitness are not distinguished, and the only information about the landscape is the number of neighbours of any given fitness that have each other fitness value. This abstraction frees us from concerns about specific local search algorithms, and enables us to address general mathematical properties.

Many real-world problems satisfy the NSF property. As an example, let us consider the Metric Travelling Salesperson Problem (TSP), whose fitness function is a sum of distances, and the number of terms in the sum is the number of cities in the TSP. The so-called 2-swap operator [3] changes only two distances in the sum for symmetric TSPs – no matter how many cities there are in the TSP. This is, in fact, the smallest change possible (in terms of the number of distances changed) while maintaining the constraint that the tour must be a Hamiltonian cycle. By ensuring that $n-2$ distances remain the same (where n is the number of cities), the 2-swap generates neighbours of which many can be considered to have similar fitness.

For many other problems, the neighbourhood operators are defined so as to change only a few terms in the sum which expresses the fitness function. For example, the maximum satisfiability problem is to find an assignment to truth variables that minimises the number of unsatisfied clauses. A clause is just a disjunction of some truth variables (or none) and some negated truth variables (or none). One way of finding a neighbour for this problem is by changing the truth value of one variable ("flipping" a variable). If the clause length is restricted to just three (variables and negated variables), and if there are 100 truth variables in the problem, then only $1 - \frac{99^3}{100^3} = 0.03$ of the clauses are likely to contain any given variable. Thus after flipping a variable 97% of the clauses will remain unchanged, and the fitness of a neighbour found by flipping a variable is likely to be similar to the original fitness.

The same argument applies to most problems whose fitness function is a sum of terms in which the number of terms increases with the number of variables in the problem. Any neighbourhood operator that changes the value of a single variable, or a small set of variables, will only change the value of a small fraction of the terms in the sum. Consequently neighbours are likely to have similar fitness. Interestingly, this characteristic is one that applies to many *PLS-complete* problems [4]. The complexity class PLS (polynomial-time local search) models the difficulty of finding a locally optimal solution to an optimisation problem. For further information on this class, we refer the interested reader to Michiels et al. [8], who list many PLS-complete optimisation problems that all have objectives which are a sum of terms which grows with problem size; among these are MaxCut with flip neighbourhood, GraphPartitioning with swap neighboudhood, MetricTSP with Lin-Kernighan neighbourhood, and CongestionGame with switch neighbourhood.

REFERENCES

[1] Kenneth De Jong. 2020. Evolutionary computation: a unified approach. In *Genetic and Evolutionary Computation Conference (GECCO) Companion*. 327–342.

[2] Benjamin Doerr and Weijie Zheng. 2020. From understanding genetic drift to a smart-restart parameter-less compact genetic algorithm. In *Genetic and Evolutionary Computation Conference (GECCO)*. ACM, 805–813.

[3] Merrill M. Flood. 1956. The Traveling-Salesman Problem. *Operations Research* 4, 1 (1956), 61–75.

[4] David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. 1988. How easy is local search? *J. Comput. System Sci.* 37, 1 (1988), 79 – 100.

[5] Andrei Lissovoi, Dirk Sudholt, Markus Wagner, and Christine Zarges. 2017. Theoretical Results on Bet-and-Run as an Initialisation Strategy. In *Genetic and Evolutionary Computation Conference (GECCO)* (Berlin, Germany) *(GECCO '17)*. ACM, 857–864.

[6] Helena R. Lourenço, Olivier C. Martin, and Thomas Stützle. 2010. Iterated Local Search: Framework and Applications. In *Handbook of Metaheuristics*. Springer, 363–397.

[7] Nicolas Meuleau and Marco Dorigo. 2002. Ant Colony Optimization and Stochastic Gradient Descent. *Artificial Life* 8, 2 (2002), 103–121.

[8] Wil Michiels, Emile H. L. Aarts, and Jan H. M. Korst. 2007. *Theoretical aspects of local search*. Springer. I–VIII, 1–235 pages.

[9] Irene Moser, Marius Gheorghita, and Aldeida Aleti. 2017. Identifying Features of Fitness Landscapes and Relating them to Problem Difficulty. *Evolutionary Computation* 25, 3 (2017), 407–437.

[10] Dale Schuurmans and Finnegan Southey. 2001. Local search characteristics of incomplete SAT procedures. *Artificial Intelligence* 132, 2 (2001), 121 – 150.

[11] Elizabeth F. et al. Wanner. 2008. Local Search with Quadratic Approximations into Memetic Algorithms for Optimization with Multiple Criteria. *Evolutionary Computation* 16, 2 (2008), 185–224.