METAHEURISTICS

FROM DESIGN TO IMPLEMENTATION

El-Ghazali Talbi
University of Lille – CNRS – INRIA

WILEY
A JOHN WILEY & SONS, INC., PUBLICATION

appearance in the set of elite solutions and the quality of solutions they have appeared.

- The usual aspiration criterion that accepts the tabu moves that generate better solutions than the best found one.

As for the intensification, the diversification of the search is not always useful. It depends on the landscape structure of the target optimization problem. For instance, if the landscape is a "massif central" where all good solutions are localized in the same region of the search space within a small distance, diversifying the search to other regions of the search space is useless. The search time assigned to the diversification and the intensification components of TS must be carefully tuned depending on the characteristics of the landscape structure associated with the problem.

TS has been successfully applied to many optimization problems. Compared to local search and simulated annealing, various search components of TS are problem specific and must be defined. The search space for TS design is much larger than for local search and simulated annealing. The degree of freedom in designing the different ingredients of TS is important. The representation associated with the tabu list, the medium-term memory, and the long-term memory must be designed according to the characteristics of the optimization problem at hand. This is not a straightforward task for some optimization problems. Moreover, TS may be very sensitive to some parameters such as the size of the tabu list.

2.6 ITERATED LOCAL SEARCH

The quality of the local optima obtained by a local search method depends on the initial solution. As we can generate local optima with high variability, iterated local search[16] may be used to improve the quality of successive local optima. This kind of strategy has been applied first in Ref. [531] and then generalized in Refs [518,726].

In *multistart local search*, the initial solution is always chosen randomly and then is unrelated to the generated local optima. ILS improves the classical multistart local search by perturbing the local optima and reconsidering them as initial solutions.

Example 2.33 Multistart local search fails for the graph bisection problem. It has been shown that many (i.e., several thousand) random initial solutions are necessary to afford stable solution quality for the graph bisection problem for instances of size $n = 500$ [420]. The number of restarts grows rapidly with the size of instances n (number of nodes) and becomes unreasonable for instances of size $n = 100{,}000$, which arises in real-life problems such as VLSI design problems.

The central limit phenomenon in the landscape of an optimization problem implies that when the size of the problem becomes very large, local optima obtained using

[16] Also known as iterated descent, large-step Markov chains, and chained local optimization.

different random initial solutions are more or less similar in terms of quality [60]. Hence, simple multistart generally fails for very large problem instances [684].

ILS is based on a simple principle that has been used in many specific heuristics such as the iterated Lin–Kernigham heuristic for the traveling salesman problem [418] and the adaptive tabu search for the quadratic assignment problem [751]. First, a local search is applied to an initial solution. Then, at each iteration, a *perturbation* of the obtained local optima is carried out. Finally, a local search is applied to the perturbed solution. The generated solution is accepted as the new current solution under some conditions. This process iterates until a given stopping criterion. Algorithm 2.10 describes the ILS algorithm.

Algorithm 2.10 Template of the iterated local search algorithm.

$s_* =$ local search(s_0) ; /* Apply a given local search algorithm */
Repeat
  $s' =$ Perturb (s_*, search history) ; /* Perturb the obtained local optima */
  $s'_* =$ Local search (s') ; /* Apply local search on the perturbed solution */
  $s_* =$ Accept (s_*, s'_*, search memory) ; /* Accepting criteria */
Until Stopping criteria
Output: Best solution found.

Three basic elements compose an ILS (Fig. 2.28):

- **Local search:** Any S-metaheuristic (deterministic or stochastic) can be used in the ILS framework such as a simple descent algorithm, a tabu search, or simulated annealing. The search procedure is treated as a black box (Fig. 2.29). In the

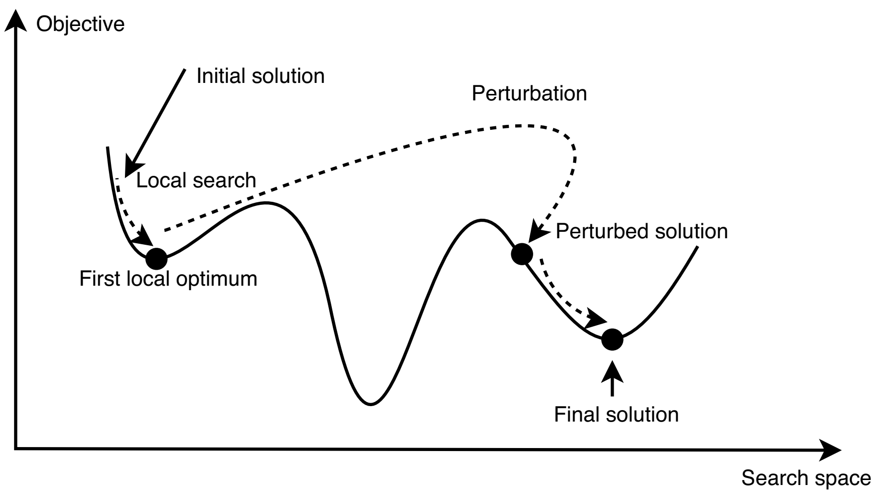

FIGURE 2.28 The principle of the iterated local search algorithm.

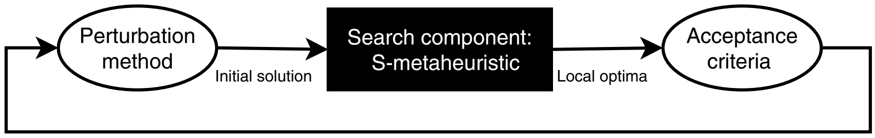

FIGURE 2.29 The search component is seen as a black box for the ILS algorithm.

literature, P-metaheuristics are not considered as candidates in the search procedure as they manipulate populations. However, some population-based metaheuristics integrate the concept of perturbation of the (sub)population to encourage the search diversification.

- **Perturbation Method.** The perturbation operator may be seen as a large random move of the current solution. The perturbation method should keep some part of the solution and perturb strongly another part of the solution to move hopefully to another basin of attraction.
- **Acceptance criteria.** The acceptance criterion defines the conditions the new local optima must satisfy to replace the current solution.

Once the S-metaheuristic involved in the ILS framework is specified, the design of ILS will depend mainly on the used perturbation method and the acceptance criterion. Many different designs may be defined according to various choices for implementing the perturbation method and the acceptance criterion.

2.6.1 Perturbation Method

The first motivation of the ILS algorithm is based on the fact that the perturbation method must be more effective than a random restart approach, where an independent initial solution is regenerated randomly. This will happen for most real-life optimization problems that are represented by structured search landscapes. For those problems, a *biased perturbation* must be carried out. However, for some problems and instances, where the landscape is rugged and flat, random restart may be more efficient. A first exercise in designing an ILS algorithm to solve the problem at hand will be to compare the performance of the random restart and the implemented biased perturbation.

A good balance must be found for the biased perturbation. The length of a perturbation may be related to the neighborhood associated with the encoding or with the number of modified components. A too small perturbation may generate cycles in the search and no gain is obtained. The probability to explore other basins of attraction will be low. Moreover, applying a local search to a small perturbation is faster than for large perturbations. Too large a perturbation will erase the information about the search memory, and then the good properties of the local optima are skipped. The extreme of this strategy is the random restart approach. The more effective is the

S-metaheuristic the larger the values of the perturbation must be. The optimal length depends mainly on the landscape structure of the optimization problem and must not exceed the correlation length (see Section 2.2.2). Even for a specific problem, it will depend on the instance at hand.

The move operator used in the perturbation may be of different nature from the neighborhood relation used in the local search procedure.

Many biased perturbation methods can be designed according to the following criteria:

- **Fixed or variable perturbations.** The length of the perturbations applied to a local optima may be defined as follows:
  - **Static.** The length is fixed *a priori* before the beginning of the ILS search.
  - **Dynamic.** The length of the perturbation is defined dynamically without taking into account the search memory.
  - **Adaptive.** In this strategy, the length of the perturbation is adapted during the search according to some informations about the search memory. Indeed, the optimal length will depend on the input instance and its structure. More information about the characteristics of the landscape may be extracted during the search.
- **Random or semideterministic perturbation.** The perturbation carried out on a solution may be a random one (memoryless) in which each move is generated randomly in the neighborhood. This leads to a Markovian chain. In semideterministic perturbations, the move is biased according to the memory of the search. For instance, the intensification and the diversification tasks of the tabu search algorithm using the medium-term memory and the long-term memory can be applied. The first approach is more popular in the literature, whereas the second approach needs more advanced and complex search mechanisms to be implemented.

2.6.2 Acceptance Criteria

The role of the acceptance criterion combined with the perturbation method is to control the classical trade-off between the intensification and the diversification tasks. The first extreme solution in terms of intensification is to accept only improving solutions in terms of the objective function (strong selection). The extreme solution in terms of diversification is to accept any solution without any regard to its quality (weak selection). Many acceptance criteria that balance the two goals may be applied:

- **Probabilistic acceptance criteria:** Many different probabilistic acceptance criteria can be found in the literature. For instance, the Boltzmann distribution of simulated annealing. In this case, a cooling schedule must be defined.

- **Deterministic acceptance criteria:** Some deterministic acceptance criteria may be inspired from the threshold accepting algorithms and the related algorithms, such as the great deluge and the record-to-record algorithms.

2.7 VARIABLE NEIGHBORHOOD SEARCH

Variable neighborhood search has been recently proposed by P. Hansen and N. Mladenovic [560]. The basic idea of VNS is to successively explore a set of predefined neighborhoods to provide a better solution. It explores either at random or systematically a set of neighborhoods to get different local optima and to escape from local optima. VNS exploits the fact that using various neighborhoods in local search may generate different local optima and that the global optima is a local optima for a given neighborhood (Fig. 2.30). Indeed, different neighborhoods generate different landscapes [428].

2.7.1 Variable Neighborhood Descent

The VNS algorithm is based on the variable neighborhood descent, which is a deterministic version of VNS. VND uses successive neighborhoods in descent to a local optimum. First, one has to define a set of neighborhood structures N_l $(l = 1, \ldots, l_{\max})$. Let N_1 be the first neighborhood to use and x the initial solution. If an improvement of the solution x in its current neighborhood $N_l(x)$ is not possible, the neighborhood structure is changed from N_l to N_{l+1}. If an improvement of the current solution x is found, the neighborhood structure returns to the first one $N_1(x)$ to restart the search (Fig. 2.31). This strategy will be effective if the different neighborhoods used are complementary in the sense that a local optima for a neighborhood N_i will not be a local optima in the neighborhood N_j. Algorithm 2.11 shows the VND algorithm.

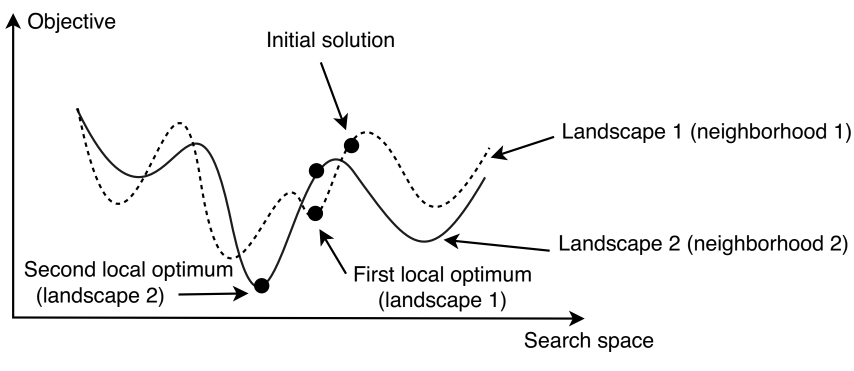

FIGURE 2.30 Variable neighborhood search using two neighborhoods. The first local optimum is obtained according to the neighborhood 1. According to the neighborhood 2, the second local optimum is obtained from the first local optimum.