7.5 Hybrid Heuristics

Ming Lu

2022-04-02

There are different ways in which we can combine the individual heuristics presented in previous sections and obtain a hybrid heuristic. Talbi in (Talbi 2002) has compiled an almost exhaustive taxonomy of hybrid heuristics, which the reader may consult as a guide for mixing heuristics according to the problem at hand. In his compilation, Talbi classifies hybrid heuristics in hierarchical or flat schemes, the latter refers to algorithms whose descriptors can be chosen in arbitrary order. But the most frequent form of hybrid heuristic is hierarchical, and is of this form that we shall limit the examples in this section. The class of hierarchical hybrid heuristics is divided in low level and high level, and within each level we may further distinguish between relay and cooperative. 11 The low level hybridization replaces a given function of a heuristic by another heuristic. In high level hybridization the different heuristics are self contained. The relay form combines heuristics in pipeline fashion: applying one after another, each using the output of the previous input. In cooperative hybridization various heuristics cooperate in parallel. The following example presents cases of high level relay hybrid heuristic. Example 7.6 Consider the basic genetic programming algorithm (Algorithm 7.4). A possible hybridization is to use a greedy heuristic to generate the initial population. This greedy strategy could consist on applying the fitness function to the initial individuals that are randomly generated, and keep those ranking high in an ordering by fitness (or discard those individuals with fitness below a certain threshold). The drawback with this extension of the GP algorithm is that it might induce over-fitted succeeding generations, and this, as it has been commented before, could trap the algorithm into a local optimum. Another hybridization is to implement a simulated annealing-like heuristic after the operation of selection in the GP algorithm, to enhance the fitness of the selected individuals but still considering as solutions some not well fitted individuals, due to the capability of making escape-moves of the SA heuristic. This is in fact an instantiation of the general probabilistic selection strategy outlined in Sect. 7.3.1, namely, to make a random selection on sorted individuals (assumed uniformly distributed) with

a probability function skewed towards best-fitted individuals. The simulated annealing implementation comes down to do this random selection under the Metropolis probability function. In this case this hybridization does provides an enhancement of the raw GP algorithm (i.e. where selection of individuals consists on simply choosing the best-fitted), for as we have commented before this random selection give theoretical guarantees of global convergence.12 Both forms presented of hybridization of GP are of the type hierarchical high level relay, since the additional heuristics (greedy in the first, SA in the second construction) are applied in addition to the operations of GP and working in tandem. Our second example presents a case of low level cooperative hybrid heuristic. These type of hybridizations are motivated by the need of improving the local search for those heuristics, such as GP or ACO, that are good in exploring the search space globally but do not care to refine, or exploit, the local findings. Example 7.7 In the ACO heuristic (Algorithm 7.6), one can refine the local search of solutions by substituting the BuildPath(k)routine by a SA heuristic, so that instead of choosing a next component solution, based on transition probabilities and pheromone values, choose a next component solution as the best solution found in a local search with SA in the neighbor of the current component solution, using as objective function the pheromone valuation, or some similar function. We leave as exercise the formalization of this hybridization.