A variable neighborhood search-based hyperheuristic for cross-domain optimization problems in CHeSC 2011 competition

Ping-Che Hsiao[1], Tsung-Che Chiang[2], Li-Chen Fu[1]

[1]Department of Computer Science and Information Engineering,
National Taiwan University, Taiwan, R.O.C.
jonasxiao@gmail.com, lichen@ntu.edu.tw
[2]Department of Computer Science and Information Engineering,
National Taiwan Normal University, Taiwan, R.O.C.
tcchiang@ieee.org

Abstract. The approach in this paper is based on Variable Neighborhood Search (VNS), which consists of two main steps: shaking and local search. Shaking disturbs solutions, and then local search seeks for the local optima. In order to adapt to different problem domains, two techniques are introduced, including dynamically adjusting strength of local search and the use of a population archive.

1 Proposed Approach

The proposed approach is based on VNS [1]. Fig. 1 shows the flow of the approach. It consists of four steps, shaking, local search, environmental selection, and periodical adjustment. Given a random initial solution, a low-level heuristic in the *mutation* and *ruin-recreate* categories is selected and applied for shaking. Then, low-level heuristics in the *local search* category are selected and applied, attempting to get to the local optima. Next, environmental selection determines which old solution in the population is replaced and selects one solution to be the base one in the next iteration. Furthermore, a periodical adjustment mechanism is introduced to adjust algorithm parameters to adapt different problem domains.

2 Shaking

In the shaking step, a low-level heuristic from *mutation* and *ruin-recreate* categories is selected and applied to the base solution x and generate a new solution x'. Here we use the tabu [2] mechanism to avoid frequently applying poor heuristics. Low-level heuristics are sorted by the amount of disturbance in ascending order. With initial tabu length equals to half of number of low-level heuristics, a heuristic is added to the tabu list (on a first-in-first-out basis) if the final solution x'' (obtained by the local search step) is worse than the starting solution x. Heuristics which generate x'' with equal fitness to x are tabued in probability of 0.2.

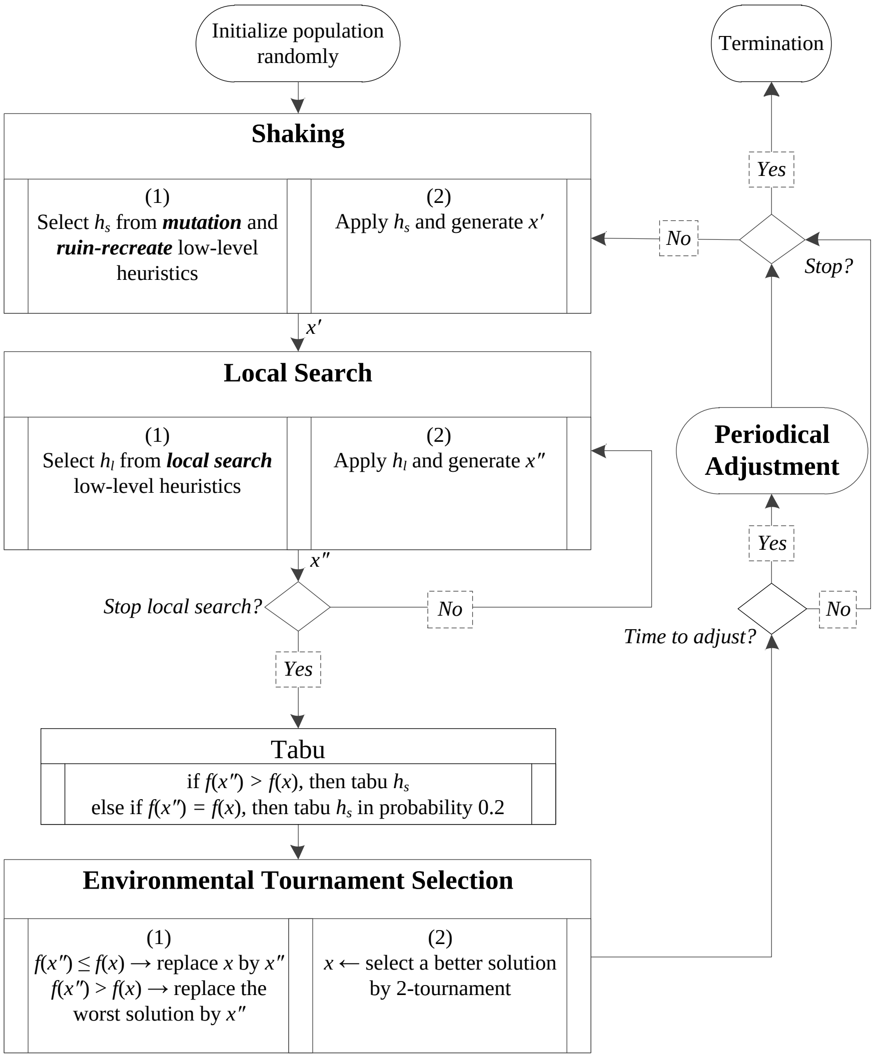

Fig. 1. Framework for the proposed approach (assuming minimization problems).

3 Local Search

In the local search step, low-level heuristics in the *local search* category are applied to the solution iteratively. We rank the heuristics by three levels:

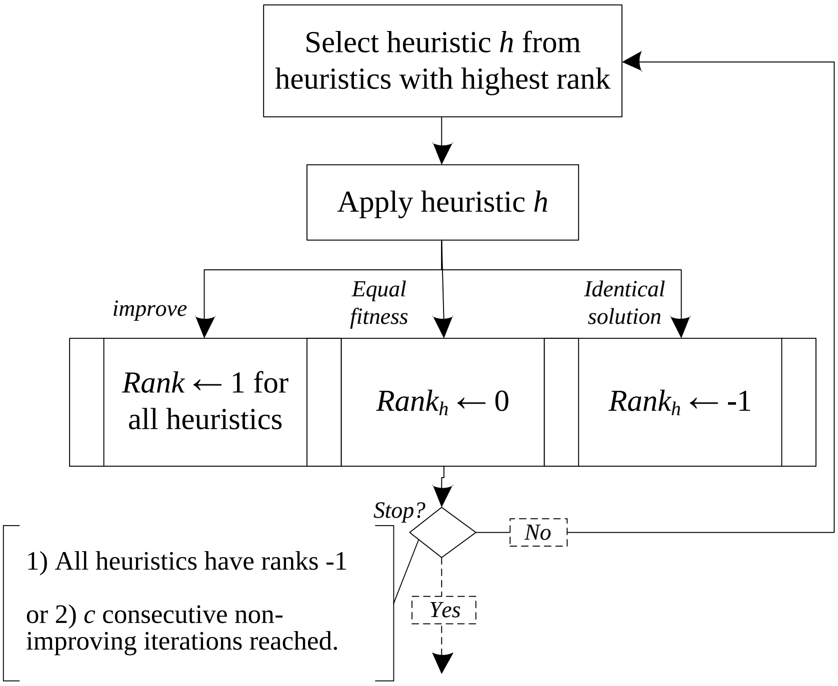

Fig. 2. Flow chart of local search step.

- Rank 1: Heuristics that are not tried or they produce a better solution in last trial.
- Rank 0: Heuristics that produce a solution with fitness equal to that of the base solution in last trial.
- Rank -1: Heuristics that produce a solution identical to the base solution in last trial.

A heuristic among those with the largest rank is selected randomly. If all heuristics have ranks -1, local search step will stop. After c consecutive non-improving iterations, local search step stops, too. The flow chart of the local search step is shown in Fig. 2. The value of c is set by 25 and may be modified in periodical adjustment step.

4 Environmental Selection

Next to the local search step is environmental selection; it includes knocking a solution out and choosing a solution to start the next iteration. When a new solution x'' is produced:

- If $f(x'')<f(x)$: Replace x by x'' in the population archive.
- Otherwise: Find a worse solution y in population by 2-tournament selection and replace y by x''.

We also use 2-tournament selection to select a better solution in the population archive as the starting solution in the next iteration.

5 Periodical Adjustment

Periodical adjustment is triggered when the algorithm consumes every 1/10 of total time budget. Two parameters will be adjusted, depending on the search progress.

The parameter *c* represents the number of consecutive non-improving iterations allowed in the local search step, and the default value of *c* is 25. Before an improving move, there may be several non-improving moves. In the periodical adjustment step, *c* is reset by the maximum number of consecutive non-improving moves which follows an improving move. To make this clear, take a look at Fig. 3. Assume we applied low-level heuristics from *local search* category for 9 times in the latest 1/10 time budget interval. A cross represents a non-improving move, and a circle represents an improving one. The maximum number of consecutive non-improving moves which follows an improving move is three in this case.

Fig. 3. Illustration of setting parameter *c*. A cross represents a non-improving move, and a circle represents an improving move.

The default population size is set by 6, and it may also be modified in the periodical adjustment step. Population is reduced to one when any of the following conditions happens:

1. Half time budget is already consumed.
2. In the latest 1/10 of time budget interval, there's no occurrence that a base solution *x* is not the best in the population but the derived new solution x'' becomes the best.

References

1. Hansen, P. & Mladenovic, N.: Variable neighborhood search: Principles and applications. European Journal of Operational Research. 130, 449--467 (2001)
2. Burke, E.K., Kendall, G. & Soubeiga, E.: A tabu-search hyperheuristic for timetabling and rostering. Journal of Heuristics. 9, 451--470 (2003)