Heuristic Problem Solving with Abstract Knowledge in the Context of the Travelling Salesperson Problem

Alexandra Kirsch

Heuristic Problem Solving with Abstract Knowledge in the Context of the Travelling Salesperson Problem

Alexandra Kirsch

Department of Computer Science, University of Tübingen

Abstract

Computationally hard problems like the Travelling Salesperson Problem, can be solved remarkably well by humans. Results obtained by computers are usually closer to the optimum, but require high computational effort and often differ from the human solutions. This paper introduces Greedy Expert Search (GES), which strives to show the same flexibility and efficiency of human solutions, while producing results of similarly high quality. A particular focus is on how to use abstract knowledge in the solution process. The Travelling Salesperson Problem serves as an example problem to illustrate and evaluate the approach.

Keywords: Problem solving, heuristics, abstraction

1. Motivation

Most of us have had some fun with automotive navigation systems. In one instance on a drive of several hours duration the satnav constantly informed us that it was replanning the route because of congestion. However, we couldn't find any change in the planned trajectory nor was there any traffic jam. We conjectured that the route was replanned at a distance of several hundred kilometres, which we would reach in the course of some hours — and thus was of no interest at all.

This is a typical example of misunderstandings between machines and users that are due to different handling of everyday problem solving tasks. Many problems that humans are confronted with, such as planning a tour through a given number of cities, are NP hard problems. Even though the space of possible solutions is huge, humans usually find remarkably good solutions. What is more, human solution strategies are stable when the problems are complicated by uncertainty, ignorance of facts or several optimisation criteria. The human ability to find satisficing [29] rather than optimal solutions seems to be at the core of this fascinating ability, or as Cassimatis [5, p. 39] states: "it may be the case that incorrectness and non-optimality are essential to human-level intelligence."

In early AI research, most notably that of Newell and Simon [26], research on artificial and natural cognition were pursued in parallel and findings from psychology were used as a basis for intelligent algorithms. This paper follows a similar approach: Greedy Expert Search (GES), the basic problem-solving strategy introduced here, is based on heuristics, a heavily used mechanism in human problem solving [**?**]. Heuristics do not only enhance efficiency: Gigerenzer and Gaissmaier [9] point out a range of real-world examples, in which heuristic problem solving leads to better results than "rational" decision-making.

Another important concept underlying intelligent behaviour is hierarchical abstraction. Typically hierarchical algorithms first find a solution in an abstracted state space, which is then refined in the original state space [19, 27]. In this paper I propose to treat abstract knowledge as a form of heuristic and show how this approach improves solutions for a specific problem, the Euclidean Travelling Salesperson Problem (TSP).

Email address: `alexandra.kirsch@uni-tuebingen.de` (Alexandra Kirsch)

A Euclidean TSP instance consists of a set of points in 2D space and symmetric path costs between them. The task is to find the shortest possible path that visits all given points. Psychological research has produced evidence that humans employ some kind of hierarchical strategy when solving Travelling Salesperson Problems [20, 11]. Explicit regions (such as membership of a city to a country) also have an influence on the human solution [36]. Similar to most of the psychological models, hierarchical search algorithms in AI typically use a strict top-down hierarchy [19, 27], where decisions on higher levels control the solution on the lower level. However, Wiener and Mallot [37] suggest a model for human wayfinding skills with a more flexible interaction of the knowledge on different levels of abstraction. In the same line, Hayes-Roth and Hayes-Roth [12] have shown that different levels of abstraction interact constantly when humans solve everyday problems.

I propose Greedy Expert Search (GES) as an algorithm to solve problems in a more human-like way. An explanation for the efficiency of human problem solving could be that we don't "compute" complete, global solutions for a problem, but proceed in a stepwise manner. Without a complete solution in mind (but possibly some sketch of the further strategy), the reaction to changes in the environment is a lot more natural than with complete global solutions.

In this paper I first introduce Greedy Expert Search (GES) as a general heuristic algorithm and then provide a set of heuristics for the TSP domain, some of which exploit abstract knowledge in the form of predefined regions. The evaluation examines on the one hand the basic heuristic search method for different TSP instances and on the other hand the benefit of abstract knowledge as heuristics versus a strict top-down approach.

This work is explicitly not intended to develop exceptionally good solutions of Euclidean TSPs (available algorithms are highly optimised and outperform humans on theoretical problem instances by far). My main objective is to suggest a flexible algorithm that can not only be applied to well-defined problems such as the TSP, but also to more realistic problems such as planning a vacation tour [31] or doing housework [2]. The reason for choosing theoretical TSP instances is that there exists a body of literature in psychology for this problem with suggestions for heuristics and there are data sets to compare the computational solution with. In addition, for more complex problems, commonsense knowledge is required, which is to some extent available in ontological knowledge bases such as Cyc [8] or ConceptNet [22], but is still not at a level to mimic human everyday knowledge. This work may also serve as a potential model for human problem solving. There is strong evidence in the domains of manipulation [13] and navigation [15, 25] that cognitive processes are controlled by heuristic mechanisms that may be modelled with GES.

2. Approach

Greedy Expert Search intends to provide a general and efficient problem-solving framework for everyday tasks. The TSP serves as a simplified example of such tasks and I use TSP-specific heuristics. The discussion at the end of the paper gives an outlook on future extensions and the potential of the method for more complex and more realistic tasks.

2.1. Greedy Expert Search (GES)

Figure 1 shows the GES algorithm in pseudocode and Figure 2 illustrates it in a schematic way. In each search step GES determines promising operators, called *horizon*, by consulting the *direction experts*. Using the successor function, the expected resulting next states are evaluated by the *evaluation experts*. The operator that is rated most highly by the evaluation experts is executed, resulting in the next state.

Beside the problem, this algorithm has as input a set of direction experts, a set of evaluation experts and combination functions for the horizons (i.e. the combination of the single results of the direction experts) and the state evaluation. The combination functions will not be considered closely in this paper. For the combination of I use horizons the union of the single horizons. The evaluation combination function is a weighted sum of the results of the evaluation experts, each returning a value indicating the quality of the proposed state (normalised between 0 and 1) and a confidence (which corresponds to the weight).

The purpose of the direction experts is to reduce the branching factor in each step. Traditionally, AI techniques consider all possible operators or when the number is very large, reduce it. GES reverses this

```
function GES  (problem, direction-experts, evaluation-experts,combine-horizon, combine-evaluations)

  function GES-get-action(s)
    if problem.goal-test(s)
      then return null
      else
        ops ← combine-horizon(map(λe: e.get-operators(s,problem.operators), e ← direction-experts))
        ops-eval ← map(λo: cons (o, combine-evaluations(map(λe: e.evaluate(o,s), e ← evaluation-experts))),
                        o ← ops)
        a ← max(ops-eval :key second)
        return a

  foreach e in direction-experts e.initialise()
  foreach e in evaluation-experts e.initialise()
  s ← problem.initial-state
  loop
    a ← GES-get-action(s)
    if a=null
      then return
      else s ← execute(a)
```

Figure 1: GES algorithm. Keywords are shown in typewriter text, functions are in italics and complex data structures (e.g. objects) are typed in bold.

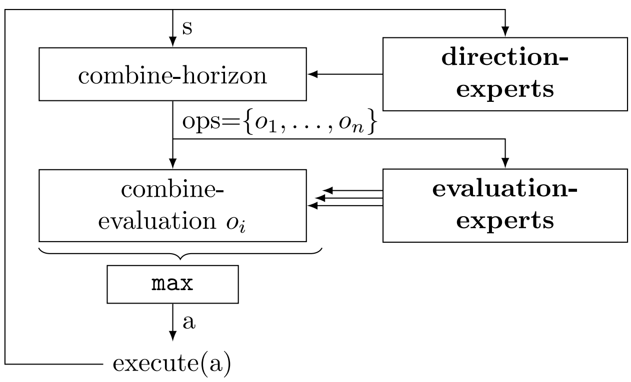
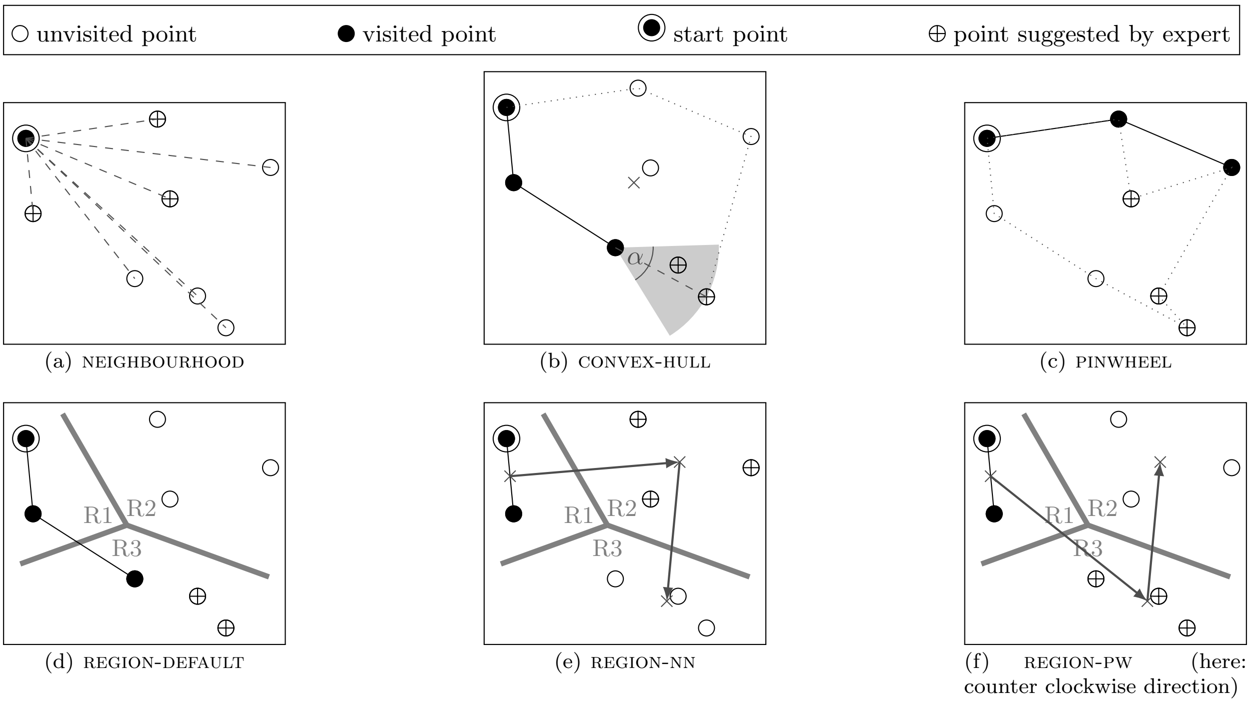


Figure 2: Schema of the GES algorithm.



(a) NEIGHBOURHOOD

(b) CONVEX-HULL

(c) PINWHEEL

(d) REGION-DEFAULT

(e) REGION-NN

(f) REGION-PW (here: counter clockwise direction)

Figure 3: Illustration of direction experts.

into a bottom-up approach by only considering operators that are explicitly suggested by a direction expert. The evaluation experts are a generalisation of heuristic functions used in other AI search methods. With the combination of all expert votes into a single numerical value for each operator in the horizon, one could model the evaluation experts together with the combination function as one single heuristic. But GES makes the single heuristics and their combination explicit.

2.2. Parametrisation for TSP Instances

For the TSP a state is a path from the given problem start point to the current decision point and the set of still unvisited points. An operator removes a point from the set of unvisited points and adds it to the path to become the new decision point.

A prerequisite for GES to be efficient is the use of efficient experts. For the TSP I used simple experts, avoiding global computations that involve the whole problem as much as possible. When global computations were unavoidable, they were mostly performed as an offline step before the search starts (this is done in the function *initialise* in the algorithm in Figure 1).

All the experts used are illustrated in Figures 3 and 4. Some of the experts contain parameters; Table 1 summarises all parameters and shows the default values used in the experiments. Most of the direction and evaluation experts are drawn from the literature on human TSP solving and I explain them along these theories.

Nearest neighbour strategy. A straightforward strategy for solving a TSP instance is to always choose the closest unvisited point. A pure nearest neighbour strategy leads to solutions that are generally worse than the solutions produced by humans [36], but it is nevertheless one aspect to consider. This idea is implemented in the NEIGHBOURHOOD direction expert, which returns the n closest unvisited points from the current end point of the tour (illustrated in Fig. 3(a) with $n = 3$). Analogously, the POINT-DISTANCE evaluation expert prefers points that are closer to the last point on the tour (Fig. 4(a)). The values are scaled so that the closest point is evaluated to 1 and the most distant point to 0.

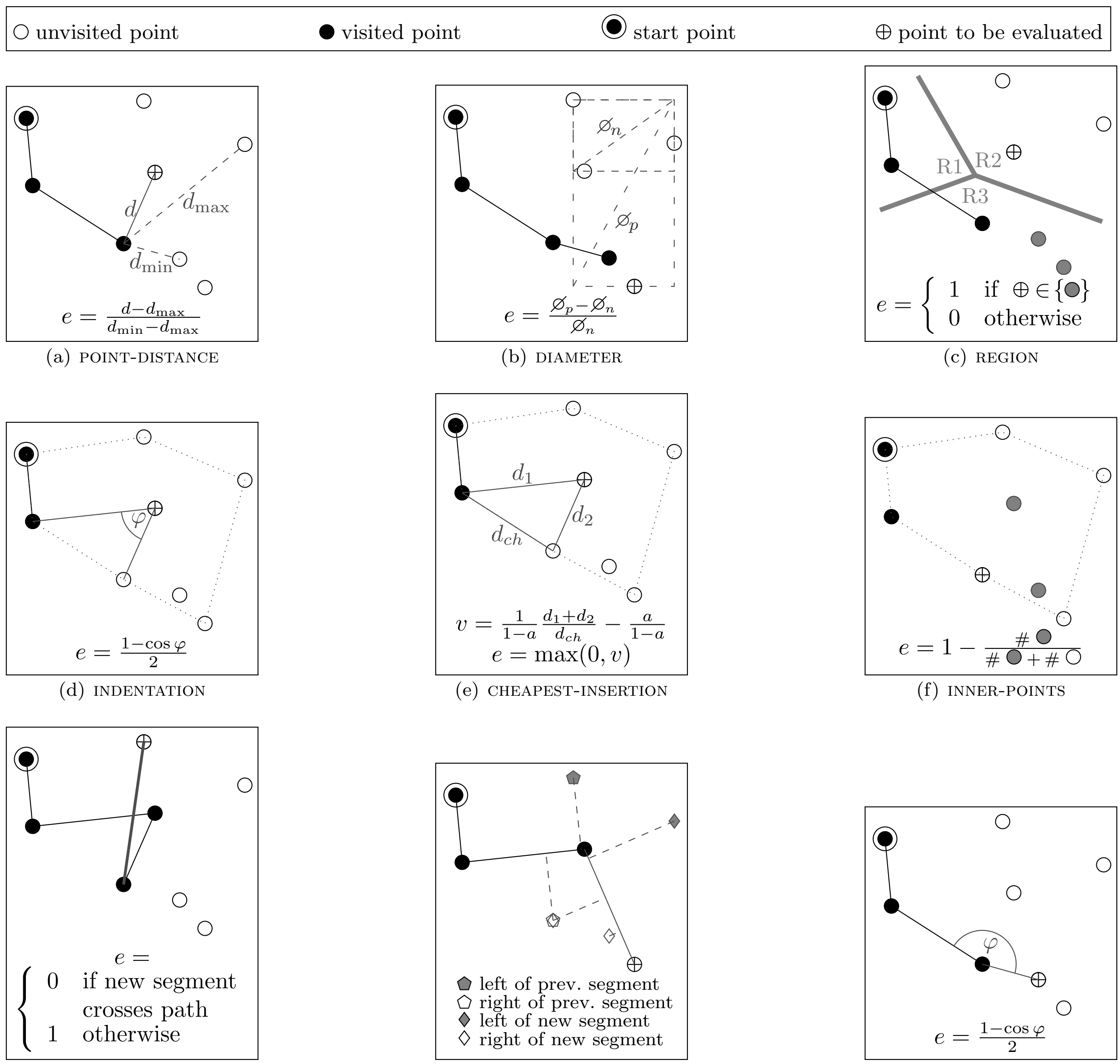


Figure 4: Illustration of evaluation experts.

Expert	Parameter	Explanation
NEIGHBOURHOOD	n	maximum number of points returned
CONVEX-HULL	$\alpha = \pi/3$	tolerance angle
PINWHEEL	$n = 3$	maximum number of points returned
PINWHEEL	$\sigma = 1$	skip tolerance: suggests points that have been "skipped" in favour of points suggested by other direction experts
PINWHEEL	direction	optional: specification of direction; if not given, the direction is determined after the first two solution steps
REGION-PW	direction	same as for pinwheel
CHEAPEST-INSERTION	$a = 1.7$	defines cutoff value for scaling (all points with $(d_1 + d_2)/d_{\mathrm{ch}} > a$ are evaluated to 0)
AVOID-SPLITTING	$\varepsilon = 0.2$	fixed value that is assigned in cases that may or may not cause the problem to be split

Table 1: Parameters used in experts. The default values in the table are used throughout the experiments, only the parameter n in the NEIGHBOURHOOD expert is varied as shown in Table 3.

Following the convex hull. In various studies, for example by Tenbrink and Wiener [32], participants have described their strategies as following a circle-like path. Based on this impression of human solution strategies and on the finding that an optimal tour follows points on the convex hull in order [10], algorithms using the convex hull have been suggested. However, global planning strategies oriented along the convex hull that include the remaining points after the contour has been computed, have been mostly dismissed by the observation that humans seem not to create the solution on a global scale, but rather construct the tour point by point [24, 34]. Some researchers have tried to combine the convex hull with a step-wise connection of points [24] similar to the approach here, while others argue that human performance can just as well be explained without the convex hull theory following a more reactive strategy [34].

The CONVEX-HULL direction expert (Fig. 3(b)) suggests the next point on the convex hull and any points lying in a segment between the current and next point on the convex hull.

The INDENTATION expert (Fig. 4(d)) is based on the assumption that humans prefer solutions with few indentations [23]. I loosen this criterion to prefer small indentations to large indentations. This is similar to the largest angle insertion heuristic [10, 24], but without any global comparison to other hypothetical insertion operations. A similar idea underlies the CHEAPEST-INSERTION expert (Fig. 4(e)). It is inspired by the cheapest insertion strategy from the literature [10, 24] but again without global comparisons. It favours the insertion of a point if the sum of the distances from the point to be considered to the last and next points on the convex hull is not much larger than the direct connection between the two points on the convex hull.

Minimising future effort. Informal observations in everyday bin packing problems provided the idea for two experts to consider the next step in light of subsequent solution steps by making the *remaining* problem as easy as possible. The DIAMETER expert (Fig. 4(b)) is based on the simple assumption that small, compact problems may be easier to solve than large, scattered problems. Therefore it favours points that reduce the diameter (determined simply by the bounding box of the problem) of the remaining problem. This is similar to the convex hull strategies, favouring points that lie on the outer rim of the problem.

In contrast, the INNER-POINTS expert (Fig. 4(f)) favours remaining problems with few inner points, leading to an early inclusion of inner points. This expert uses the convex hull to define inner points as all points not lying on the convex hull and follows the observation that problems with many inner points are difficult to solve for humans [23, 35].

$\exists$⬟	T	T	F	F	F	F	F	T	T	F	F	F	T	T	T	T
$\exists$⬠	F	F	T	T	F	F	F	F	F	T	T	F	T	T	T	T
$\exists$◆	T	F	F	F	T	F	F	T	F	T	T	T	T	T	F	F
$\exists$◇	F	F	T	F	F	T	F	T	T	T	F	T	T	F	T	F
	$e = 1$							$e = \varepsilon$					$c = 0$			

Table 2: Rules for deciding whether a problem is/will be split with the insertion of the next line. The symbols are explained in Figure 4(h), and $\exists$ is an abbreviation of whether any such points exists.

Pinwheel strategy. Also striving for a circular solution, the PINWHEEL direction expert computes the centre of mass of the TSP instance and then suggests the points as they lie on a projected circle around this centre (Fig. 3(c)). With a combination of a pinwheel and nearest neighbour strategy Best and Simon [4] could explain human solution behaviour to a large extent.

Avoiding intersections. All studies have shown that the solutions of humans seldom contain intersecting lines. This is not surprising as optimal solutions of TSPs cannot contain intersections [10]. There is an ongoing debate whether people avoid intersections deliberately or if the results are intersection-free because they are usually near the optimum [34].

The AVOID-INTERSECTIONS evaluation expert (Fig. 4(g)) rates a point to be inserted with 0 if its connecting line intersects the existing path and with 1 if it does not. In most instances, however, this expert makes no difference, because intersections are usually caused by earlier decisions and at the moment when the intersection is produced there are only few or no alternatives left that would lead to an intersection-free path.

A more indirect way of avoiding intersections is to avoid lines that "split" the problem, i.e. to avoid situations in which large parts of the remaining problem lie on both sides of the partial solution. A general solution of whether a problem is split by a line would require a sophisticated pattern matching algorithm. For the AVOID-SPLITTING expert (Fig. 4(h)) I used a simplified solution that counts the number of points lying to the left and right of the last segment of the current state and the points on the left and right of the line that will potentially be added and then uses a set of simple rules (listed in Table 2) to determine whether the resulting state would be a split problem. In some cases, the points indicate a split, but it may be resolved in the next solution step. These cases are evaluated by a fixed value between 0 and 1 as given by the parameter ε.

Visual considerations. The AVOID-SPLITTING expert is also an example that visual properties of the planar TSP may play a role for humans to solve them. Pizlo et al. [28] even propose a purely image-based approach to model human TSP solving and achieve impressive results. And Vickers et al. [35] have shown that the aesthetic form of a trajectory is somehow equivalent to good TSP solutions.

However, real-world versions of the TSP are usually not given in the form of a map, but human solutions are still comparable to abstract versions [36]. Therefore, I have only included simple visual information in the experts. The FOLLOW-LINES evaluation expert (Fig 4(i)) prefers the continuation of straight lines as a simple approximation to aesthetically pleasing paths.

Hierarchical strategies. Several hierarchical solution approaches for the TSP have been suggested, which were also inspired and compared to human problem solving skills. Kong and Schunn [20] describe a classical 'coarse-to-fine' approach: first a set of regions is defined using k-means clustering. The centroids of the clusters define a more abstract problem, which is solved first. The points of the original problem are then added along the global path. For this kind of approach Best [3] has proposed to use a distorted model for the distances between points, based on observations that humans underestimate intra-cluster distances and overestimate inter-cluster distances, which shows that standard clustering is not necessarily the best approach for structuring TSP instances. Pizlo et al. [28] present another hierarchical approach, in which the hierarchy is based on perceptual properties and the number of layers is determined by the algorithm. Also in this case the lower levels are bound to the solutions of the higher levels.

In GES, such abstract knowledge is treated like any other heuristic, thus providing evidence for or against a decision, but never completely restricting the options on the lowest level of abstraction. As pointed out before, identifying visual patterns is not the focus of this paper. Therefore, in the problems used here, the points annotated with explicit assignments to a region (see evaluation section for the different ways used to define the regions). Wiener et al. [36] also used predefined regions in their studies and predefined regions exist in real-world problems for example in the forms of countries.

Three direction experts use the given information about regions: region-default, region-nn and region-pw (Figures 3(d)– 3(f)). All three experts return all the points in the same region as the last point of the current partial tour. When no points in the region are left, region-default returns no points; region-nn returns all the points of the region that is closest to the last region, measured by the centre of mass of each region; region-pw returns all the points of the region that follows in a circular shape (as in the pinwheel expert, but using the regions' centres of mass instead of problem points). All these experts can be combined with other direction experts. region-default can be combined with the neighbourhood expert in a way that when region-default returns no points, the suggestions of neighbourhood are used. This mode is very similar to the region-nn expert, only that the next region is determined on the lower level with closeness of points, whereas region-nn uses the proximity of the region. In the evaluation, only the region-pw is used, because all three direction experts using regions lead to similar results.

Information about regions can also be used in evaluation experts. The region expert (Fig. 4(c)) favours candidate points that lie in the same region as the last point on the path.

3. Evaluation

The purpose of this work is to sketch a general, heuristics-based problem-solving mechanism that can eventually assist and advice people in real-world problems. The Travelling Salesperson Problem serves as an example, but it is not the purpose of this work to solve this particular problem. The evaluation can therefore only show tendencies of strengths and weaknesses of GES and its applicability to problems like the TSP. A particular emphasis lies on the use of abstract knowledge in the form of a heuristic in contrast to strictly hierarchically organised top-down processes.

I first explain the experimental configurations for solving TSPs and the data sets used in the evaluation.

3.1. Experimental Configurations

I use six sets of configurations of GES composed of the experts described above, summarised in Table 3. In each set 5–8 configurations are defined along a certain pattern and each corresponds to configurations in the other sets except for the set-specific feature.

std These configurations do not use direction experts (this is emulated by configuring the neighbourhood to return all unvisited points) and thus could be emulated by a standard greedy search that combines the evaluation experts as a weighted sum into one heuristic value. *std-NN* represents the standard nearest neighbour strategy. Even though it is well-known that humans use more sophisticated strategies, it serves as a baseline. *std-1*–*std-5* use different combinations of the available evaluation experts except the region expert. *std-CH-1* and *std-CH-2* use primarily experts that make use of the convex hull.

These configurations were selected based on informal experimentation and to show a variety of configurations. Automatic optimisation is difficult for TSPs, because every instance is different and there are no classification schemes available. Besides, the intention of this evaluation is primarily to demonstrate the strengths and weaknesses of the approach rather than obtaining optimal TSP solutions.

ges These are the same configurations as the *std* set, except for the direction experts. The *ges* configurations restrict the horizon by using direction experts that give only a subset of the unvisited points. For the two configurations that are based primarily on convex hull strategies, the horizon is the union of points returned by the convex-hull and neighbourhood experts. For all other configurations, the pinwheel direction expert is used instead of convex-hull.

	direction experts						evaluation experts								
	NEIGHBOURHOOD	CONVEX-HULL	PINWHEEL	REGION-DEFAULT	REGION-NN	REGION-PW	POINT-DISTANCE	INDENTATION	INNER-POINTS	CHEAPEST-INSERTION	PROBLEM-DIAMETER	AVOID-INTERSECTIONS	FOLLOW-LINES	AVOID-SPLITTING	REGIONS
std: Standard search only using evaluation experts															
std-NN	$n = 21$						1								
std-1	$n = 21$						1				.5	.5	.5	.5	
std-2	$n = 21$						1				1	.5	1	.5	
std-3	$n = 21$						.2				1	.5	.5	.5	
std-4	$n = 21$						1				.5	.5			
std-5	$n = 21$						1	1					1	.5	
std-CH-1	$n = 21$						1	1	1						
std-CH-2	$n = 21$						1	0.5		0.5					
ges: Search with restricted branching factor by direction experts															
ges-CH-*	$n = 3$	✓													
ges-*	$n = 3$		✓												
ges-reg: Region information in evaluation experts															
ges-CH-reg-*	$n = 3$	✓													1
ges-reg-*	$n = 3$		✓												1
hr: Region information in direction experts															
hr-{1...5}	$n = 3$					✓									
hr-reg: Region information in direction and evaluation experts															
hr-reg-{1...5}	$n = 3$					✓									1
td: Emulation of top-down strategy															
td-{1...5}						✓									

Table 3: Used combinations of experts. The numbers in the evaluation expert fields indicate the confidences (weights). Fields marked in grey take the values of the respective entries from the std configurations (for example hr-2 uses the same evaluation experts as std-2).

ges-reg Identical to *ges*, except for the addition of the REGION evaluation expert with weight 1.

hr Identical to the respective *ges-1*–*ges-5* configurations, except that the PINWHEEL direction expert is replaced by the REGION-PINWHEEL direction expert.

hr-reg Identical to *hr*, except for the addition of the REGION evaluation expert with weight 1.

td Identical to the *hr* configurations, but without the NEIGHBOURHOOD direction experts. This emulates a strict top-down approach, where the regions restrict the solution options on the lower level (i.e. there is a global strategy based on regions). Adding the REGION evaluation expert to these configurations does not change the results, because all points under consideration as returned by the REGION-PINWHEEL expert lie in the same region anyway and thus all receive the same evaluation value from the REGION expert.

3.2. Data Sets

The primary data set used in the evaluation was collected in the form of an online game. In this game[1] players have to solve different TSP instances and are rewarded with scores relative to the tour length. The data set is openly available[2] and is updated automatically every night. The data used here includes all played games up to 25 July, 2014, comprising a total of 2115 tours by 54 players. The game has 24 levels, which are organised in three blocks:

Levels 1–8 standard TSP instances with a given set of points, the starting point is not given. Since the GES approach needs a starting point, the point that happens to be first in the TSP specification file is used for the generated solutions.

Levels 9–16 standard TSP instances, but with a predefined starting point;

Levels 17–24 TSPs with region information by coloured points. The players are merely informed that points in these levels have different colours and that this should not distract them.

Some TSP instances are used in several level blocks to provide the possibility to better compare the conditions. The problems themselves are partly taken from data sets from the literature [23], own previous studies [17, 18], generated randomly or constructed to examine the use of clusters and regions (here I do not analyse the human strategies in detail, but only use the tour lengths as comparison). The problems contain between 5 and 21 points.

As the game data set only contains 8 problems with region information, another data set of 10 randomly generated problems with 10 points each was used. The regions were either defined by clustering or points were randomly assigned to regions. For clustering the Expectation-Maximisation (EM) algorithm as implemented in the WEKA data mining software [38] was used and each problem was divided into two, three and four clusters. In all, for each randomly generated problem, there were four different variants of region assignments.

3.3. Properties of GES for TSP Solving

For examining the properties of GES when solving TSPs with the experts used here, I used the data set from the TSP game. I address the following aspects:

1. quality of the generated solutions,
2. impact of restricting the branching factor by using direction experts,
3. impact of the specific choice of expert combinations in general.

The standard measure for TSP solution quality is the percentage above the optimal tour length (PAO). Figure 5 shows the average, minimum and maximum PAO value of all *std* and *ges* configurations respectively and as a comparison the values of human solutions over all the 24 levels of the TSP game. In most studies on human TSP solving, the participants have only one trial and the GES approach can currently also produce only one solution. This is why Figure 5 differentiates between the first trial of each participant and all trials that include repetitions of the same level.

The first trials of the participants show smaller variation and are on average slightly better than all trials. It seems that for some difficult levels, after some suboptimal trials, players venture more creative solutions that are often longer than their first attempts. Therefore, the first trials seem to be a better comparison for the GES solutions and will be used in the following comparisons.

The minimum and average human solutions are in general better than the ones produced by the configurations of GES, but in some instances the algorithm can compete with the average or even minimum solution. The worst results produced by GES are, however, shorter than the worst results by the players. This may to some extent be explicable to noise in the data of the TSP game, but it is confirmed by previous

[1] `www.perlentaucher.medieninformatik.uni-tuebingen.de`

[2] `http://www.wsi.uni-tuebingen.de/lehrstuehle/human-computer-interaction/home/code-datasets/tsp-dataset.html`

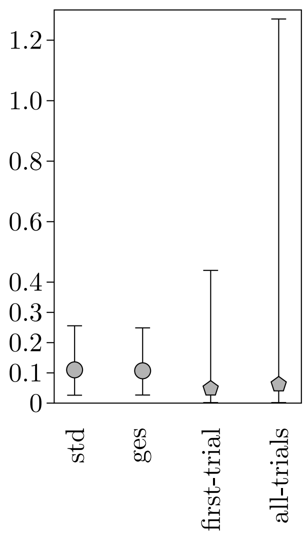


Figure 5: Average PAO (given as decimal numbers, 1.0 equals 100%) over all levels 1–24 for the two sets of configurations that ignore regions and the players of the game. The whiskers show the minimum and maximum PAO values, averaged over the shown problem instances.

Problem	**std-1**	**ges-1**	**std-2**	**ges-2**	**std-3**	**ges-3**	**std-4**	**ges-4**	**std-5**	**ges-5**	**first-trial**
Level 1	0.00	0.00	0.00	0.00	0.36	0.36	0.00	0.00	0.00	0.00	0.07
Level 2	0.10	0.10	0.10	0.10	0.01	0.01	0.10	0.10	0.00	0.00	0.01
Level 3	0.11	0.11	0.12	0.12	0.04	0.04	0.11	0.11	0.03	0.03	0.07
Level 4	0.01	0.01	0.01	0.01	0.38	0.38	0.05	0.05	0.02	0.02	0.03
Level 5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02
Level 6	0.10	0.10	0.10	0.10	0.24	0.24	0.16	0.16	0.13	0.13	0.06
Level 7	0.14	0.14	0.14	0.14	0.20	0.14	0.05	0.05	0.12	0.12	0.06
Level 8	0.11	0.11	0.18	0.18	0.26	0.18	0.14	0.14	0.25	0.10	0.05
Level 9	0.08	0.08	0.08	0.08	0.20	0.20	0.12	0.12	0.14	0.14	0.03
Level 10	0.06	0.06	0.06	0.06	0.04	0.04	0.06	0.06	0.05	0.05	0.05
Level 11	0.06	0.06	0.02	0.02	0.31	0.31	0.06	0.06	0.01	0.01	0.02
Level 12	0.30	0.30	0.01	0.01	0.28	0.01	0.11	0.11	0.04	0.04	0.07
Level 13	0.21	0.21	0.09	0.09	0.05	0.05	0.21	0.21	0.18	0.18	0.06
Level 14	0.12	0.12	0.07	0.08	0.16	0.13	0.10	0.10	0.10	0.10	0.11
Level 15	0.02	0.02	0.02	0.02	0.02	0.02	0.12	0.12	0.38	0.38	0.04
Level 16	0.15	0.27	0.14	0.14	0.40	0.40	0.00	0.00	0.41	0.41	0.02
Level 17	0.16	0.16	0.16	0.16	0.09	0.09	0.06	0.06	0.09	0.09	0.02
Level 18	0.30	0.30	0.12	0.12	0.24	0.06	0.24	0.24	0.13	0.11	0.05
Level 19	0.10	0.10	0.10	0.10	0.24	0.24	0.16	0.16	0.13	0.13	0.04
Level 20	0.00	0.00	0.10	0.10	0.10	0.10	0.13	0.13	0.15	0.07	0.01
Level 21	0.11	0.11	0.18	0.18	0.26	0.18	0.14	0.14	0.25	0.10	0.05
Level 22	0.02	0.02	0.02	0.02	0.02	0.02	0.12	0.12	0.38	0.38	0.07
Level 23	0.20	0.20	0.20	0.20	0.29	0.29	0.27	0.27	0.33	0.33	0.02
Level 24	0.15	0.27	0.14	0.14	0.40	0.40	0.00	0.00	0.41	0.41	0.04

Table 4: Comparison of *std* search configurations to *ges* configurations that use direction experts, with comparison to average PAO values of human first trials in the TSP game.

results with data sets that were obtained in more controlled studies [17]. This is no guarantee that no GES configuration could produce worse results than humans in their first trial, but it gives some indication about the stability of combining several heuristics, which is by no means self-evident.

Table 4 confirms these observations for the single levels and the single expert combinations for the mixed configurations as compared to the average first-trial results. The table also shows the influence of restricting the branching factor by direction experts (for the *-nn* and *CH-** combinations, the results are identical in the pairs of *std* and *ges* configurations). It is not surprising that in many cases, there is no difference, and in three instances the tour gets longer (from the view of the algorithm, it is only two instances, because levels 16 and 24 have the same layout, but level 24 additionally has a random assignment of colourings/regions, which is not used in the *std* and *ges* configurations). So a reasonable limitation of the branching factor by heuristics does not significantly threaten the outcome and it may be suspected that this will also hold for other problems than the TSP, provided the direction experts are adequately chosen.

What is surprising in Table 4, is that in 10 instances (9 when accounting for identical layouts of levels 8 and 21) the tours get shorter. This happens in two of the configurations and may thus only happen in combination with specific evaluation experts. The direction experts are a first filter of possible next steps and thus can forestall the consideration of very bad options. Appropriate evaluation experts could reject such bad options, but because the judgements of all evaluation experts are combined in a weighted sum, an exclusion of some options up front seems to be rather beneficial than deteriorating. A similar effect could also be obtained by changing the combination of the judgements of evaluation experts, e.g. by some kind of veto than can exclude an option from further consideration.

On the whole, the use of direction experts is not only a promising way to increase efficiency, but also to help guide the search into promising directions. Because the results of the *std* combinations are so similar to the *ges* configurations, only include the *ges* configurations are included in the subsequent evaluations.

The third aspect to discuss is the stability of solution quality with respect to different expert combinations. So how much does GES depend on being configured with the perfect combination of experts? This is a very important question with respect to its scalability to more experts, its generality for other problems and its general usefulness from a software engineering perspective. As mentioned before, there is no obvious classification scheme for TSP instances — as for many real-world problems — and thus configurations should be able to cope with a wide range of instances. Figure 6 shows the average, minimum and maximum results for different expert configurations for the TSP instances of levels 17–24 (as only those contain region information, which is necessary for all combinations shown except *ges*).

The basic *ges* configurations are rather stable with respect to the specific experts and their weights. However, when the REGION expert is added, the specific configuration matters a lot more. One possible reason is the relative influence of this evaluation expert, which is always weighted with 1, with respect to the total of the other experts. This can explain the worsening of results in configuration *ges-reg-4*, where the sum of weights of the other experts is 2.0. But for *ges-reg-3*, this explanation is not sufficient, since the other weights with 2.7 are almost as influential as in *ges-reg-1* with 3.0. The single addition of the REGION expert seems to make the whole composition of experts more brittle. In contrast, the use of the *region-pw* direction expert does not destroy the stability with respect to evaluation expert combinations, while still making use of region information.

The general design principle employed in creating expert combinations seems to be at least as important as the specific weights as demonstrated by the generally bad solutions of the *td* configurations and the constant good performance of the *ges* and *hr* configurations. So as a guideline for further usage, the expert combinations should follow some problem-specific general design principles and the exact configurations should be determined empirically. But — at least for the TSP problems here — the overall performance is not considerably weakened by the choice of the exact parameters.

3.4. Hierarchical Knowledge in GES

In a heuristic decision-making framework such as GES, abstract knowledge such as the regions in TSP instances, can be integrated both as a classical top-down process by using the direction experts to restrict the space of possible next steps as implemented in the *td* configurations, or as an additional heuristic to guide

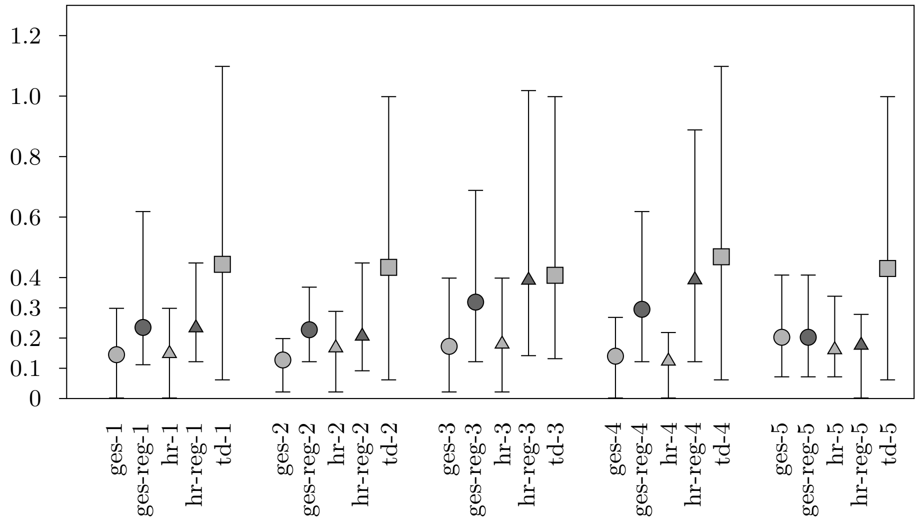


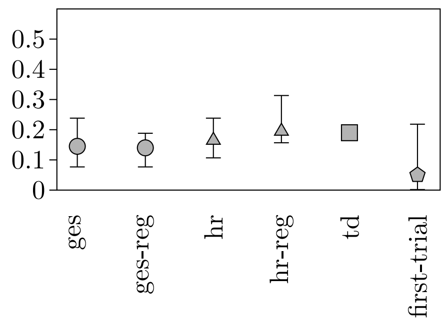
Figure 6: Influence of specific composition of experts for different sets of configurations (levels 17–24).

the search, without fully determining it, by using evaluation experts as in the *ges-reg* configurations, by using direction experts as in the *hr* configurations, or by a combination of both as in the *hr-reg* configurations.

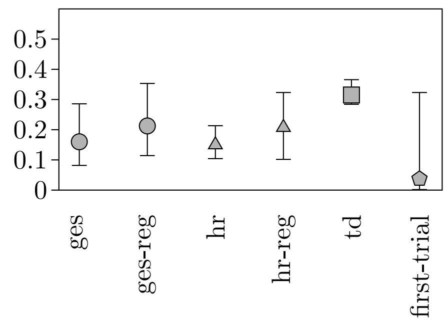
Figure 6 shows that both the *hr* and *td* configurations are rather stable with respect to the combination and weights of the evaluation experts, whereas *hr-reg* and *ges-reg* suffer from some dependence on the exact evaluation expert combination. Figure 7 gives a more detailed view on the different types of problems: Figure 7(a) shows the performance of each set of configurations for the two levels with region assignments, where the regions have to be followed in order to find the optimal path (of course, not every path that follows the regions is also optimal). All the configurations show a similar level of performance and consistent with the findings shown before, are worse than human performance, but still in a similar range, with average PAO values between the average and worst human performance in the first trial. Figure 7(b) shows the performance for levels where visible clusters are emphasised by the region colouring, but strictly following these regions does not result in the optimal solution. These levels were also harder for the humans, at least the worst results were further from the optimum as for the first levels with helpful region assignments. The configurations using the REGION evaluation expert produce slightly worse results than their counterparts without this expert and the *td* configurations rather tend towards the worst human results than towards their average as the other configurations do.

The real challenge comes with those levels where points are assigned randomly to regions (Figure 7(c)). The *hr* configurations are here similarly stable as the *ges* configurations (which ignore the regions). Because of the combination with another direction expert, the useless region information is practically discarded, even though the algorithm can profit from it when the regions are meaningful. The configurations with the additional REGION evaluation expert get more distracted by the random regions, but by far not as much as the *td* configurations. Again, none of the configurations can compete with human results, but *ges* and *hr* produce results between the average and worst human solution, *ges-reg* and *hr-reg* produce on average solutions with the length of the longest human tours, whereas even the best solutions produced by the *td* configurations are much longer than the worst human results.

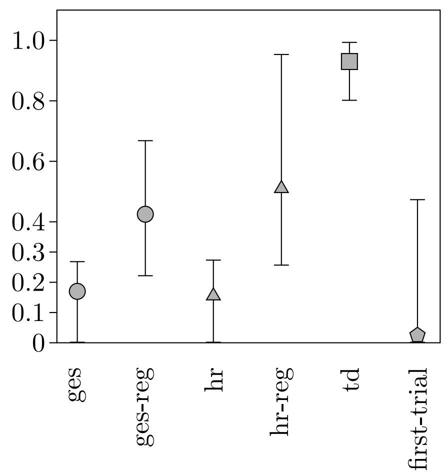
These observations are confirmed when using the data set with 10 randomly generated problems (Figure 8). In Figure 8(a) for each algorithm the best solution that could be found with any of the predefined clusterings (of two, three and four clusters) was used for the evaluation. This corresponds to the ideal situation that a clustering algorithm would return the clustering that is most convenient for the specific problem and configuration. Even though following the clusters does not necessarily lead to the optimal solution, all the configurations show a similar quality of solutions. However, when the regions are composed of random points (Figure 8(b)), the picture is analog to Figure 7(c).



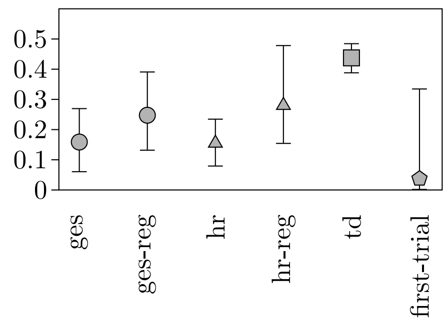
(a) Levels with helpful region assignments (18, 21)



(b) Levels with clusters that lead to a suboptimal solution (17, 19, 22, 23)

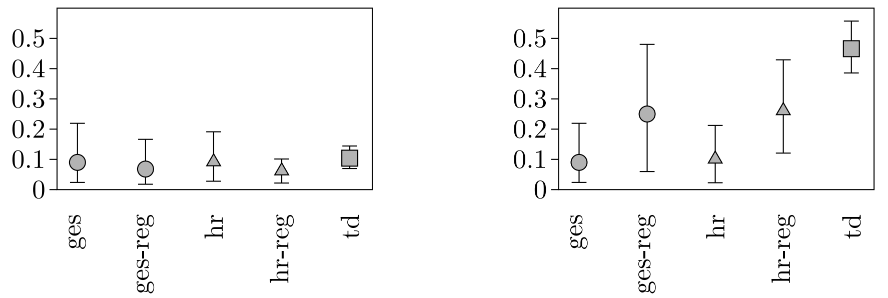


(c) Levels with random region assignments (20, 24)



(d) All levels.

Figure 7: Average PAO for different configurations and human performance on first trial for levels 17–24. The whiskers show the minimum and maximum PAO values, averaged over the shown problem instances.



(a) Regions according to clusters (with best clustering per problem and configuration).

(b) Random assignment of points to regions.

Figure 8: Average PAO for different configurations with randomly generated data set with given regions.

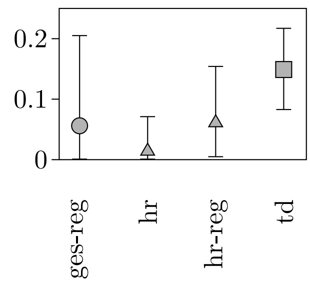
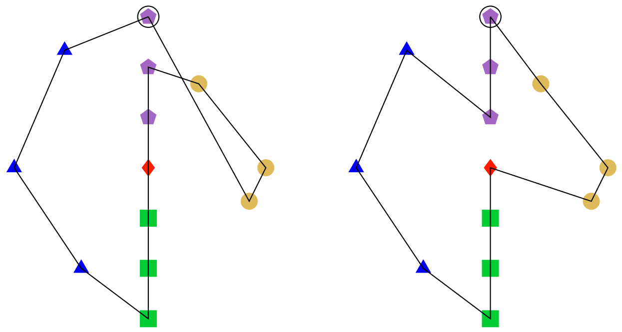


Figure 9: Dependence of configurations on clustering of points: average difference in PAO when clustering points into two, three or four regions. The whiskers show the average of minimum and maximum difference over all ten problems.

One can argue that a random assignment of points to regions is a rather extreme situation and that the whole sense of the regions is to provide some abstract spatial information. But this situation can occur to a lesser degree, for example when cities are assigned to administrative regions, the knowledge of regions does not necessarily help to make correct decisions (for example, most people estimate San Diego, California, to lie west of Reno, Nevada, because California is west of Nevada [30], but actually Reno is further west than San Diego). There is also the problem that clusters as defined by point distances do not necessarily provide a good structure for a TSP instance,for example when the centres of all clusters are in a line. Figure 9 shows the average, minimum and maximum differences between the best and worst solution when clustering with two, three or four clusters in the random problem data set. While the *hr* configurations are hardly affected by the number of clusters, the *td* configurations show a much more pronounced difference.

The evaluations shown so far with aggregated PAO values indicate that information about regions should better be ignored altogether (the *ges* configurations mostly produce the best tours, only equalled by the *hr* configurations, where one can argue that the good results are due to the NEIGHBOURHOOD direction expert rather than the REGION-PW expert). Figure 10 shows a rather difficult TSP instance, where the regions lead to the optimal solution. While it is almost impossible to find the optimum with the "flat" experts used here, the REGIONS expert leads to a much better solution.

In several other instances, the use of regions led to a better solution, but this advantage is counteracted by results from other instances where regions are less helpful. Some adaptation of expert configurations to specific problem instances would therefore be desirable. Since this choice of configuration will always be difficult, it is still desirable that any expert configuration is at least to some degree stable towards different problem instances.



(a) ges-CH-1: 10% above the optimum

(b) ges-CH-1-reg: 4% above the optimum

Figure 10: Comparison of solutions of *ges-CH-1* configuration with and without the use of regions for level 21.

4. Discussion

Relation to Human Problem Solving. Tenbrink and Wiener [32] have found that "TSP-related strategies have hitherto often been treated as real alternatives that are mutually exclusive. Our results indicate that they may be better represented as a repertory of strategies and subprocesses that are available to humans when solving TSP tasks. The relative weight of each particular subprocess or strategy may differ substantially between individuals and subtasks."[32] GES formalises this idea, not specifically for the TSP, but for any kind of problem solving task. Tenbrink and Seifert [31] asked participants to plan a holiday trip and analysed verbal reports from this task. They found that humans combined spatial knowledge with knowledge about travel modalities, for example the physical effort when riding a bike. This behaviour can be reproduced by GES with a combination of experts containing different kinds of knowledge.

Research on human problem solving and creativity usually differentiates between *divergent* and *convergent thinking*. Divergent thinking refers to the generation of several alternative solutions, whereas convergent thinking is the decision process to produce a final solution out of a number of alternatives [1]. Whereas people's creativity is often limited by a lack of divergent thinking, AI techniques typically consider *all* possible options, thus diverging too much. GES explicitly includes a divergence step with the direction experts and a convergence step with the evaluation experts.

Wiener and Mallot [37] have studied human route planning. They have suggested a 'fine-to-coarse' strategy that takes into account region knowledge in path planning, but unlike 'coarse-to-fine' strategies that promote a strict top-down approach, they suggest that in working memory a simplified copy of the hierarchical organisation is held and modified during the planning process. In this paradigm the membership of a point to a region can be adapted according to the situation. Although in the presented work the regions are fixed, a similar effect occurs by treating regions as knowledge that can potentially be adapted in the problem solving process.

Related Work in AI. The GES algorithm used here is most related to the decision-making procedure in the cognitive architecture FORR [6]. Instead of differentiating between direction and evaluation experts, FORR classifies heuristics into *perfect heuristics*, *planners* and *ordinary heuristics* and these classes are consulted sequentially. GES can also include planners, but in the generic form of direction and evaluation experts.

The IBM Watson program, which beat expert human players in the game of Jeopardy, uses a similar approach as GES with experts providing candidates for answers and evaluating such candidates in parallel [7]. Watson additionally uses machine learning to determine the weights of experts. The scope of the Watson project is a lot larger than the work presented here and the focus was more on outperforming humans than to imitate them.

In AI hierarchical knowledge is usually exploited in hierarchical, top-down algorithms to enhance search efficiency. Knoblock [19] has shown that hierarchical decomposition returns optimal solutions if the subproblems are independent, but that the solution quality is still acceptable when this assumption is slightly violated, as it is in most real-world problems. The abstraction strategies in AI planning and abstraction systems, in which states are clustered into state sets, have been analysed by Pang and Holte [27].

In hierarchical A* search [14] solutions of abstracted versions of the problem are used as heuristics for the underlying flat A* search. The use of the abstract search result is similar to the method proposed here to use hierarchical information as additional knowledge. However, the underlying search paradigms are very different: while A* uses one heuristic function, GES is designed for a wide variety of knowledge sources without requiring heuristics to explicitly estimate expected path costs and explicitly generates operators, while discarding the goal of reaching optimal solutions. Leighton et al. [21] propose a hierarchical A* strategy to trade off search quality against computational efficiency, in accordance with the traditional view with the goal of optimal solutions, which is only loosened for practical applicability.

Rapidly-exploring random trees (RRT) are often used for problems that require high reactivity such as robot navigation. RRTs determine new states to explore by sampling from a probability distribution. This is somehow similar to the direction experts of GES, but the process is rather implicitly defined by the probability distribution than explicitly by heuristics. Urmson and Simmons [33] propose several variations of the basic RRT algorithm to tweak this expansion process, also by including heuristics, but still the process is primarily a sampling process.

Outlook. This paper has presented results of solving TSP instances with GES. The greedy approach prohibits combinatorial explosion and the direction experts limit the branching factor, thus GES is by construction very efficient (of course, the efficiency also depends on the experts used). The stepwise construction of the solution also allows for dynamics in more realistic application domains.

Some more deliberate processes are necessary for more complex problems. But rather than including them as a fixed component, GES offers to include them in the form of direction or evaluation experts. In the TSP domain, a clustering algorithm could take the role of a planner. The experiments show that for the TSP such abstract information cannot always be trusted and the combination of different experts helps to overcome inadequate higher-level decisions.

The results indicate that the combination of experts is rather stable in the setting used here and thus may scale well to more experts and other problems. This, however, has to be verified in other problem domains. In addition, GES should be enhanced with an automatic learning mechanism to adapt the expert configurations to problem instances, for example by using the Robot Learning Language [16] to automatically collect experiences and continually adapt the program.

The basic principle of GES is to combine different pieces of problem-related knowledge rather than to optimise a certain variable or set of variables. Even though literature from psychology and neuroscience has promoted this approach for years, the majority of AI researchers still adhere to the optimisation principle. Heuristic approaches may look less formal and less controllable, and one cannot easily provide guarantees of runtime or solution quality. But real-world problems are just not as well-defined as we would like them to be and once we accept heuristic methods as a viable approach, theoretical frameworks may emerge for a better understanding of these solutions. This paper proposes a small step into the direction of heuristics-based problem solving methods that may provide better solutions to everyday problems than the optimisation algorithms used today.

Acknowledgements

With the support of the Bavarian Academy of Sciences and Humanities.

References

[1] Arden, R., Chavez, R. S., Grazioplene, R., Jung, R. E., 2010. Neuroimaging creativity: A psychometric view. Behavioural Brain Research 214 (2), 143–156.

[2] Beetz, M., Bandouch, J., Kirsch, A., Maldonado, A., Müller, A., Rusu, R. B., 2007. The assistive kitchen — a demonstration scenario for cognitive technical systems. In: Proceedings of the 4th COE Workshop on Human Adaptive Mechatronics (HAM).

[3] Best, B. J., 2006. Cognitive approaches to the traveling salesperson problem: Perceptual complexity that produces computational simplicity. In: Papers from the American Artificial Intelligence Association (AAAI) Spring Symposium.

[4] Best, B. J., Simon, H. A., 2000. Simulating human performance on the traveling salesman problem. In: Proceedings of the Third International Conference on Cognitive Modeling.

[5] Cassimatis, N. L., 2012. Human-level artificial intelligence must be an extraordinary science. Advances in Cognitive Systems 1, 37–45.

[6] Epstein, S., 2004. Metaknowledge for autonomous systems. In: AAAI Spring Symposium on Knowledge Representation and Ontology for Autonomous Systems.

[7] Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A. A., Lally, A., Murdock, J. W., Nyberg, E., Prager, J., Schlaefer, N., Welty, C., 2010. Building watson: An overview of the deepqa project. AI Magazine.

[8] Foxvog, D., 2010. Cyc. Springer, pp. 259–278.

[9] Gigerenzer, G., Gaissmaier, W., 2011. Heuristic decision making. Annual Review of Psychology 62, 451–482.

[10] Golden, B., Bodin, L., Doyle, T., Stewart, W., 1980. Approximate travelling salesman algorithms. Operations Research 28, 694–711.

[11] Graham, S. M., Joshi, A., Pizlo, Z., 2000. The traveling salesman problem: A hierarchical model. Memory & Cognition 28 (7), 1191–1204.

[12] Hayes-Roth, B., Hayes-Roth, F., 1979. A cognitive model of planning. Cognitive Science 3 (4), 275–310.

[13] Herbort, O., Butz, M. V., May 2012. The continuous end-state comfort effect: weighted integration of multiple biases. Psychological Research 76 (3), 345–363.

[14] Holte, R., Perez, M. B., Zimmer, R. M., Macdonald, A. J., 1996. Hierarchical A*: Searching abstraction hierarchies efficiently. In: Proceedings of the National Conference on Artificial Intelligence.

[15] Huber, M., Su, Y.-H., Krüger, M., Faschian, K., Glasauer, S., Hermsdörfer, J., February 2014. Adjustments of speed and path when avoiding collisions with another pedestrian. PLoS ONE 9 (2).

[16] Kirsch, A., 2009. Robot learning language — integrating programming and learning for cognitive systems. Robotics and Autonomous Systems Journal 57 (9), 943–954.
URL http://dx.doi.org/10.1016/j.robot.2009.05.001

[17] Kirsch, A., 2011. Humanlike problem solving in the context of the traveling salesperson problem. In: AAAI Fall Symposium on Advances in Cognitive Systems.

[18] Kirsch, A., 2012. Hierarchical knowledge for heuristic problem solving — a case study on the traveling salesperson problem. In: First Annual Conference on Advances in Cognitive Systems.

[19] Knoblock, C. A., 1991. Search reduction in hierarchical problem solving. In: Ninth International Conference on Artificial Intelligence. AAAI Press.

[20] Kong, X., Schunn, C. D., 2007. Global vs. local information processing in visual/spatial problem solving: The case of traveling salesman problem. Cognitive Systems Research 8, 192–207.

[21] Leighton, M. J., Ruml, W., Holte, R. C., 2011. Faster optimal and suboptimal hierarchical search. In: Proc. 4th Symposium on Combinatorial Search (SoCS).

[22] Liu, H., Singh, P., 2004. ConceptNet: A Practical Commonsense Reasoning Toolkit. BT Technology Journal 22, 211–226.

[23] MacGregor, J., Ormerod, T., 1996. Human performance on the traveling salesman problem. Perception & Psychophysics 58 (4), 527–539.

[24] MacGregor, J., Ormerod, T., Chronicle, E., 2000. A model of human performance on the travelling salesperson problem. Memory & Cognition 7, 1183–1190.

[25] Moussaïd, M., Helbing, D., Theraulaz, G., 2011. How simple rules determine pedestrian behavior and crowd disasters. Proceedings of the National Academy of Sciences of the United States of America 108 (17), 6884–6888.

[26] Newell, A., Simon, H., 1972. Human Problem Solving. Prentice Hall, Upper Saddle River, New Jersey.

[27] Pang, B., Holte, R. C., 2011. State-set search. In: Proc. 4th Symposium on Combinatorial Search (SoCS).

[28] Pizlo, Z., Stefanov, E., Saalweachter, John; Li, Z., Haxhimusa, Y., , Kropatsch, W. G., 2006. Traveling salesman problem: A foveating pyramid model. The Journal of Problem Solving 1 (1).

[29] Simon, H. A., 1956. Rational choice and the structure of the environment. Psychological Review 63 (2), 129–138.

[30] Stevens, A., Coupe, P., 1978. Distortions in judged spatial relations. Cognitive Psychology 10, 526–550.

[31] Tenbrink, T., Seifert, I., 2011. Conceptual layers and strategies in tour planning. Cognitive Processing 12, 109–125.

[32] Tenbrink, T., Wiener, J., 2009. The verbalization of multiple strategies in a variant of the traveling salesperson problem. Cognitive Processing 10, 143–161.

[33] Urmson, C., Simmons, R., October 2003. Approaches for heuristically biasing rrt growth. In: IEEE/RSJ IROS 2003.

[34] Vickers, D., Lee, M., Dry, M., Hughes, P., 2003. The roles of the convex hull and the number of potential intersections in performance on visually presented traveling salesperson problems. Memory & Cognition 31 (7), 1094–1104.

[35] Vickers, D., Lee, M. D., Dry, M., Hughes, P., McMahon, J. A., 2006. The aesthetic appeal of minimal structures: Judging the attractiveness of solutions to traveling salesperson problems. Perception & Psychophysics 68 (1), 32–42.

[36] Wiener, J., Ehbauer, N., Mallot, H., 2009. Planning paths to multiple targets: memory involvement and planning heuristics in spatial problem solving. Psychological Research 73, 644–658.

[37] Wiener, J. M., Mallot, H. A., 2003. 'Fine-to-Coarse' route planning and navigation in regionalized environments. Spatial Cognition and Computation 3 (4), 331–358.

[38] Witten, I. H., Frank, E., Hall, M. A., 2011. Data Mining: Practical Machine Learning Tools and Techniques, 3rd Edition. Morgan Kaufmann.