A Multi-Objective Ant Colony Approach for Pareto-Optimization Using Dynamic Programming

Sascha Häckel
Department of Economics and Business Administration
Chemnitz University of Technology
Chemnitz, Germany
shae@hrz.tu-chemnitz.de

Marco Fischer
Department of Economics and Business Administration
Chemnitz University of Technology
Chemnitz, Germany
mfi@hrz.tu-chemnitz.de

David Zechel
Department of Economics and Business Administration
Chemnitz University of Technology
Chemnitz, Germany
zed@hrz.tu-chemnitz.de

Tobias Teich
Department of Economics and Business Administration
Zwickau University of Applied Science of West Saxony
Zwickau, Germany
tobias.teich@fh-zwickau.de

ABSTRACT

This paper covers a multi-objective Ant Colony Optimization, which is applied to the $\mathcal{NP}$-complete multi-objective shortest path problem in order to approximate Pareto-fronts. The efficient single-objective solvability of the problem is used to improve the results of the ant algorithm significantly. A dynamic program is developed which generates local heuristic values on the edges of the problem graph. These heuristic values are used by the artificial ants.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods, Dynamic programming*

General Terms

Algorithms, Design, Measurement, Verification

Keywords

Multi-Objective Optimization, Hybridization, Ant Colony Optimization, Dynamic Programming, Shortest-Path Problems, Pareto-Optimization

1. INTRODUCTION

Shortest path problems and traveling salesman problems are typical problems with a broad application range in practice. Some problems belong to complexity class $\mathcal{P}$ and may be solved efficiently within appropriate time with the help of known algorithms. Other mostly $\mathcal{NP}$-complete problems turn out to be more difficult and, therefore, require the use of heuristic methods such as Ant Colony Optimization in order to determine approximative solutions. There are some problems that can be assigned to complexity class $\mathcal{P}$ by applying a single-criterion objective function. The multi-objective version of the problem is $\mathcal{NP}$-complete. We are going to exemplarily use the multi-objective shortest path problem in directed, non-cyclical graphs. The single-objective version of this problem belongs to problems in complexity class $\mathcal{P}$ and can be solved efficiently. DIJKSTRA developed an algorithm that solves the problem in a complexity of $\mathcal{O}\left(n^2\right)$ [4]. The algorithm of WARSHALL and FLOYD even specifies all shortest paths between all nodes in $\mathcal{O}\left(n^3\right)$ with the help of dynamic programming (see in [16], [9]).

The multi-objective version of the shortest path problem is $\mathcal{NP}$-complete and even the bi-objective case is intractable (see in [11], [15]). Especially for big problem instances, it is impossible to use an algorithm, which definitely solves the problem. Therefore, a heuristic method is used to solve the Problem, in this case a multi-objective Ant Colony Optimization approach. This article shows how to use the single-objective solvability to improve the multi-objective ant algorithm.

2. PROBLEM DESCRIPTION

The multi-objective shortest path problem considered here is based on a non-cyclical, directed graph (digraph) D, which can be described as the tuple in equation (2.1).

$$D = \{V, E\} \tag{2.1}$$

The sets V and E are to be described as following in (2.2).

$$\begin{aligned} V &= \{v_i : i = 1, ..., z;\ z \in \mathbb{N}\} \\ E &= \{e_{ij} : v_i \rightarrow v_j : v_i, v_j \in V;\ v_i \neq v_j\} \\ & V \neq \emptyset,\ E \subseteq V^2 \end{aligned} \tag{2.2}$$

The set V with its elements $v_i \in V$ describes all nodes of the graph. These nodes are connected by edges $e_{ij} \in E$. An edge e_{ij} depicts a directed relation between the nodes v_i and v_j. The node v_i is called start node of the edge and v_j is called end node of the edge. The direction of an edge can be described using the notation $e_{ij} = v_i \rightarrow v_j$. The non-cyclical character of the graphs used here is specified by the condition in equation (2.3).

$$\begin{aligned} \nexists C &= \{(v_k \rightarrow v_{k_1}), (v_{k_1} \rightarrow v_{k_2}), ..., \\ & \quad (v_{k_{l-1}} \rightarrow v_{k_l})(v_{k_l} \rightarrow v_k)\} \subseteq E \end{aligned} \tag{2.3}$$

Besides the structure of the graph there is the fact that edges have weights. Every weight describes the separate cost or the use of an edge in terms of a component for the later to be defined objective function for a criterion. Due to the multi-objective nature of the problem a separate weighting function for each edge and for all criteria is defined. The function f_e^c in equation (2.4) describes the weighting function for the edges of the graph having criterion c. The number of all criteria of the problem is described by n.

$$\begin{aligned} f_e^c : e_{ij} \mapsto \mathbb{R}^+\ \forall e_{ij} \in E \quad \text{with} \quad c = 1, \ldots, n \\ 2 \leq n \in \mathbb{N} \end{aligned} \tag{2.4}$$

The paths $\pi(v_s, v_t)$ through the graph between two defined nodes v_s and v_t are solutions of the problem. The set of all possible solutions of the problem is described by the solution space $\Pi(v_s, v_t)$ [see equation (2.5)]. The nodes v_s and v_t are called start and end node of the path.

$$\begin{aligned} \Pi(v_s, v_t) = \{\pi(v_s, v_t) : \{(v_s \rightarrow v_{s+1}), (v_{s+1} \rightarrow v_{s+2}), \\ \ldots, (v_{t-1} \rightarrow v_t)\} \subseteq E\}\} \end{aligned} \tag{2.5}$$

Every path is rated regarding all criteria with a aggregation of the weighting functions for each edge of the path. Our research project examines a problem with three independent or conflicting criteria which are subject to different aggregation methods ($n = 3$).[1] The first criterion equals a monetary value of cost, which is the result of summing up all weights along the certain path. The second criterion is an aggregated probabilistic value. The particular weights of the edges describe independent occurrence probabilities of local events. The aggregated probability value of a path results from the product of all weights along the path. The third criterion is subject to the MIN-MAX aggregation method. Therefore, the objective function value is created from the maximum of all weights along the certain path.

While the criteria $c = 1$ and $c = 2$ are in a conflicting relation, criterion $c = 3$ can be identified as independent.

The calculation used here can be formalized by function f_π^c in equation (2.6).

$$f_\pi^c(\pi(s,t)) = \begin{cases} \sum\limits_{\forall e_{ij} \in \pi(s,t)} f_e^c(e_{ij}) & \text{if } c = 1 \\ \prod\limits_{\forall e_{ij} \in \pi(s,t)} f_e^c(e_{ij}) & \text{if } c = 2 \\ \max\limits_{\forall e_{ij} \in \pi(s,t)} f_e^c(e_{ij}) & \text{if } c = 3 \end{cases} \tag{2.6}$$

Expecting that all criteria should be minimized (criteria that should be maximized have to be inverted properly), the objective function of the problem can be described as F in equation (2.7) (see vector optimization in [13]).

$$\min \ F(\pi) = \begin{pmatrix} f_\pi^1 \\ f_\pi^2 \\ \vdots \\ f_\pi^c \\ \vdots \\ f_\pi^n \end{pmatrix} \quad \text{with } \pi \in \Pi \tag{2.7}$$

It is rarely possible that a single *perfect* solution π^{**} exists that is optimal regarding all criteria and dominates all other solutions. As we have an existing conflict between two criteria in our example, there is no possibility for a perfect solution [14]. That is why the set of all efficient solutions should be determined. The efficiency of the solutions results from the non-dominance and the Pareto-optimality of the solution, respectively. A solution π_A dominates the solution π_B weakly ($\pi_A \succeq \pi_B$) if at least the condition in equation (2.8) is effective [3].

$$\begin{aligned} f_\pi^c(\pi_A) \leq f_\pi^c(\pi_B) \quad \forall c \in \{1, \ldots, n\} \\ f_\pi^c(\pi_A) < f_\pi^c(\pi_B) \quad \exists c \in \{1, \ldots, n\} \end{aligned} \tag{2.8}$$

A solution π_A is regarded as efficient, if there is no solution $\pi_B \in \Pi$ which dominates π_A weakly. [see equation (2.9)].

$$\pi_A \in \Pi^* \Leftrightarrow \nexists \pi_B \in \Pi : \pi_B \succeq \pi_A \tag{2.9}$$

These solutions $\pi_A \in \Pi^*$ are also called Pareto-optimal [3]. In the target area the objective function values of the solutions in Π^* set up the so called Pareto-front. $\Pi^* \subseteq \Pi$ describes the set of all efficient (Pareto-optimal) solutions. The subject of the optimization process is to obtain Π^* and therefore, all efficient solutions.

3. MULTI-OBJECTIVE ANT COLONY OPTIMIZATION

To solve the described optimization problem, due to the $\mathcal{NP}$-completeness the Ant Colony Optimization (ACO) Metaheuristic is applied. The subject of the algorithm is a Pareto-optimization to determine all non-dominated solutions of the problem.

3.1 Ant System

Initial point is the Ant System (AS)[2], an algorithm developed by DORIGO, MANIEZZO and COLORNI [6]. This is where artificial ants are used to determine the shortest path within a graph. Each ant moves through the graph and

[1]The assumption $n = 3$ is only an example and can be extended to any number of criteria.

[2]originally Ant System / ant-cycle

constructs a path π_k. The ant has to decide locally which edge to choose at each node. This random decision by using the so called *random proportional rule* is made based on all available edges ($\mathcal{N}_i$) that go off from the local node position v_i [see equation 3.1)].

$$p(e_{ij},t)=\begin{cases}\frac{(\tau_{ij}(t))^{\alpha}\cdot(\eta_{ij})^{\beta}}{\sum\limits_{j\in\mathcal{N}_i}(\tau_{ij}(t))^{\alpha}\cdot(\eta_{ij})^{\beta}} & \text{if} \quad j\in\mathcal{N}_i\\ \\ 0 & \text{otherwise}\end{cases} \tag{3.1}$$

The heuristic information η_{ij} and the pheromone information $\tau_{ij}(t)$ of an edge e_{ij} at the specific moment t between the nodes v_i and v_j influence the decision. The heuristic information depicts a local information about the length of a section or the attractiveness of a single alternative, respectively. The pheromone information includes a global memory of the ants and therewith an experience about the global attractiveness of alternatives. For the Traveling Salesman Problem (TSP) the set $\mathcal{N}_i$ of all available alternatives is built from the total quantity of edges that go out from node v_i without the edges leading to already visited nodes. Using the shortest path problem in non-cyclical digraphs the node v_t has to be reachable through all edges of the set $\mathcal{N}_i$. Therefore, all edges that do not lead directly or indirectly to the node v_t ($e_{ij}\notin$ **feasible**) are removed from the graph.[3] Algorithm 1 shows the possibility of how to mark all feasible edges in the graph. Therewith the subset relation $\mathcal{N}_i\subseteq$ **feasible** arises.

Algorithm 1 Marking all feasible edges (and nodes) in the graph for the solution construction of ACO

```
function: DetermineFeasibleEdges
requires digraph D, node $v_t$
begin
 1: feasible ← feasible ∪ $v_t$
 2: for all edges $e_{ij} \in E : j = t$ do
 3:    feasible ← feasible ∪ $e_{ij}$
 4:    if node $v_i \notin$ feasible then
 5:       DetermineFeasibleEdges(D, $v_i$)
 6:    end if
 7: end for
end function
```

The parameters α and β in equation (3.1) regulate the influence of the pheromone and heuristic information. After each solution construction the pheromone information is updated based on equation (3.2). All of the edges used by the ants in the iteration are incremented. After each iteration an evaporation across all edges of the graph that is affected by parameter ρ is accomplished.

$$\tau_{ij}(t+1)=\rho\cdot\tau_{ij}(t)+_{\Delta}\,\tau_{ij}(t,t+1) \tag{3.2}$$

The value ${}_{\Delta}\tau_{ij}(t,t+1)$ is calculated using equation (3.3).

$$\begin{aligned}\Delta\tau_{ij}(t,t+1)&=\sum_{k=1}^{m}\Delta\tau_{ij}^{k}(t,t+1)\\ \Delta\tau_{ij}^{k}(t,t+1)&=\begin{cases}\frac{Q}{L^k(t)} & e_{ij}\in\pi_k\\ 0 & \text{otherwise}\end{cases}\end{aligned} \tag{3.3}$$

[3]This results from the fact that we use a non-cyclical digraph.

Q depicts a constant and has to be defined as a parameter. $L^k(t)$ equals the length or the cost of the path which has been constructed by the ant k in iteration t.

3.2 Bi-Objective Pareto-Optimization Using Ant System

IREDI, MERKLE and MIDDENDORF introduced a bi-objective version of Ant System for Pareto-optimizing [12]. In this version m ants are divided onto w colonies. The colonies get heterogeneous weighting regarding the optimization criteria. The ants within colonies also get heterogeneous weights. The weighting parameter $\lambda\in[0,1]$ describes the relative importance of the first criterion. The weight of the second criterion results to $(1-\lambda)$. The assignment of weighting intervals to the colonies can be done disjoint or overlapping. In case the intervals are disjoint there is no possibility that two ants have identical weights. The colonies have intervals separated from each other which are distributed to the respective ants. The weighting parameter λ^k of an ant $k\in[1,m/w]$ from colony $g\in[1,w]$ is determined by equation (3.4).

$$\lambda^k=(g-1)\cdot m/w+k \tag{3.4}$$

The overlapping approach not considered here has weighting intervals of the colonies that overlap by 50%, so that at least two ants with the same weight exist.

An ant chooses an edge e_{ij} at node v_i from the alternative set $\mathcal{N}_i$ with probability p which is shown in equation (3.5). The pheromone and the heuristic information for the first criterion are described by the symbols τ_{ij} and η_{ij}. The symbols τ'_{ij} and η'_{ij} correspond to the pheromone and heuristic information for the second criterion.

$$p(e_{ij})=\frac{\tau_{ij}^{\lambda\alpha}\cdot\eta_{ij}^{\lambda\beta}\cdot\tau_{ij}^{\prime(1-\lambda)\alpha}\cdot\eta_{ij}^{\prime(1-\lambda)\beta}}{\sum_{h\in N_i}\tau_{ih}^{\lambda\alpha}\cdot\eta_{ih}^{\lambda\beta}\cdot\tau_{ih}^{\prime(1-\lambda)\alpha}\cdot\eta_{ih}^{\prime(1-\lambda)\beta}} \tag{3.5}$$

After finishing the solution construction of all ants, the pheromone matrices are updated. Two matrices per colony are used, one for each criterion. In this context, multiple strategies are considered which ants may use for updating. Each colony g manages a local front, which is derived from the set Π_g^{bsf}. This set includes all recently found *best-so-far* solutions that are not dominated by any other solution recently found by any ant of the colony g. Furthermore a global front which includes all *best-so-far* solutions (Π^{bsf}) of all ants of all colonies is managed.

Depending on the strategy, an ant may either perform an update if it found a solution at the local or at the global front. Experiments in [12] showed that the update strategy of the global front achieved better results, which is why this strategy is pursued more specifically.

The pheromone update of an ant k to the related edges $e_{ij}\in\pi_k$ will be performed with the value ${}_{\Delta}\tau_{ij}^{k}$, determined by equation (3.6).[4] The set K_t contains all ants eligible for updates of the iteration t.

$${}_{\Delta}\tau_{ij}^{k}(t,t+1)=\frac{1}{|K_t|} \tag{3.6}$$

In this paper we use the update strategy by origin of the ants. This is where the pheromone matrices of the colony the ant belongs to are updated.

[4]The notation $|K_t|$ describes the length of set K_t (number of elements).

3.3 Optimizing More than Two Objectives

The bi-objective approach we described is to be extended to $n > 2$ criteria. We replace m by a configurable sampling parameter $\chi_{\text{ants}} > 1$ which determines the granularity of the weights between the criteria. Therefore, the number of ants m automatically results from χ_{ants}. The weights of the criteria are captured in a weighting vector $\vec{\lambda_k}$, which is assigned injectively to ant k. We also define the set Λ^n_{ants}, which includes all feasible weighting vectors for the ants of an n-criteria problem. Λ^n_{ants} is defined in equation (3.7).

$$\Lambda^n_{\text{ants}} = \left\{ \vec{\lambda_k} = \begin{pmatrix} \lambda^1_k \\ \vdots \\ \lambda^c_k \\ \vdots \\ \lambda^n_k \end{pmatrix} : \begin{array}{l} \lambda^c_k \in \{0, \frac{1}{\chi_{\text{ants}}}, \frac{2}{\chi_{\text{ants}}}, \ldots, 1\}; \\ \sum_{c=1}^{n} \lambda^c_k = 1; \\ \forall k = 1, \ldots, m \end{array} \right\} \tag{3.7}$$

The number of ants m results from parameter n and χ_{ants} and can be determined by equation (3.8).

$$m(n, \chi_{\text{ants}}) = \begin{cases} \chi_{\text{ants}} & \text{if } n = 2 \\ \sum\limits_{i=1}^{\chi_{\text{ants}}} m(n-1, i) & \text{otherwise} \end{cases} \tag{3.8}$$

Equal to IREDI ET AL. the ants are pooled to colonies. The sampling parameter χ_{colonies} handles the generation of the weighting vectors $\vec{\lambda_g}$ for the colonies $g = 1, \ldots, w$.[5] To avoid empty colonies the condition $\chi_{\text{colonies}} \leq \chi_{\text{ants}}$ needs to be complied. An ant k will be assigned to colony g if the vector distance from $\vec{\lambda_k}$ to $\vec{\lambda_g}$ is the shortest. Formally, this relation can be described by equation (3.9).[6]

$$k \in g : \arg \min_{\forall g=1,\ldots,w} \sqrt{\sum_{c=1}^{n} \left(\lambda^c_k - \lambda^c_g\right)^2} \tag{3.9}$$

The calculation of probability p according to how the ant makes its decision will be changed to equation (3.10).

$$p\left(e_{ij}\right) = \frac{\prod_{c=1}^{n} (\tau^c_{ij})^{\alpha \cdot \lambda^k_c} \cdot (\eta^c_{ij})^{\beta \cdot \lambda^k_c}}{\sum_{N_i} \prod_{c=1}^{n} (\tau^c_{ij})^{\alpha \cdot \lambda^k_c} \cdot (\eta^c_{ij})^{\beta \cdot \lambda^k_c}} \tag{3.10}$$

Towards the original approach we only use one pheromone matrix for all criteria.

3.4 Extensions to Ant Colony System

In the following the approach described in the last section will be combined with the main properties of Ant Colony System (ACS) [5]. The decision rule for choosing an edge e_{ij} from the set of alternatives ($\mathcal{N}_i$) is extended by parameter $q_0 \in [0, 1]$. q_0 describes the probability that an ant selects the edge with the best value $p\left(e_{ij}\right)$ from the set $\mathcal{N}_i$. Otherwise, a decision is made using the *Random Proportional Rule* as already described (J).

Formally, this so called *Pseudo-Random Proportional Rule* can be described by equation (3.11).

$$e_{ij} = \begin{cases} \arg\max\limits_{l \in N_i} p\left(e_{il}\right) & \text{if} \quad q \leq q0 \\ J & \text{otherwise} \end{cases} \tag{3.11}$$

The global pheromone update is accommodated to the concept of Ant Colony System according to equation (3.12). The update quantity ${}_{\Delta}\tau_{ij}$ is still be determined as seen in equation (3.6). The evaporation will only be applied on edges with a positive update quantity.

$$\begin{gathered} \tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \rho \cdot_{\Delta} \tau_{ij}(t, t+1) \\ \forall \ \Delta\tau_{ij}(t, t+1) > 0 \end{gathered} \tag{3.12}$$

Furthermore, in ACS a local pheromone update is introduced. Each edge e_{ij} selected in an ant's local decision experiences a negative update. So the attractiveness of those edges is lowered in comparison to the alternatives and a higher exploration is targeted. The negative update is realized through equation (3.13). $\xi \in [0, 1]$ affects the intensity of the local negative update.

$$\tau_{ij} = (1-\xi) \cdot \tau_{ij} + \xi \cdot \tau_0 \tag{3.13}$$

The parameter τ_0 describes the initial value of the pheromone information.

4. HEURISTIC INFORMATION

The determination of the heuristic information values η_{ij} has not been addressed so far and should be described subsequently. In the first stage of the optimization run via Ant Colony Optimization the most important issue is to lead the artificial ants into promising regions of the solution space that will be searched through more intensely in later stages. But at the beginning there are no significant pheromone information. To avoid a purely stochastic search in the first iterations, the heuristic information values are used to control the exploration. Therefore, the heuristic information values are essential for creating good initial solutions with a high level of diversity and affect the algorithm behavior significantly.

In the following, we discuss the heuristic information values used in the standard case. After that, we will introduce an alternative concept based on Dynamic Programming for the problem considered here.

4.1 Inverted Edge Weights

In the standard case the heuristic information values η_{ij} for shortest path problems or TSP are based on the local edge weights of the graph [10]. As shown in equation (4.1), the inverted local edge weights for minimization problems are used as heuristic information η^c_{ij} of edge e_{ij} for criterion c.

In the following we refer to this approach as Standard Heuristic Information.

$$\eta^c_{ij} = \frac{1}{f^c_e\left(e_{ij}\right)} \tag{4.1}$$

This concept corresponds with a simple Greedy heuristic and is not really suitable to create good initial solutions, especially for multi-objective problems.

Because of the strict local principle, edges with extremely varying objective function values $f^c_e\left(e_{ij}\right)$ (either very good

[5] w is also determined as m by equation (3.8) using χ_{colonies} instead of χ_{ants}.

[6] In this case the colonies are disjoint.

or very bad) depict a major problem. The probability distribution for local decisions in the first iterations are already extremely high or low for single edges. There is a risk that entire parts of the graph are excluded from the search process. Huge areas of the solution space may possibly not be reached using this concept. This could lead to a early convergence of the algorithm to a local optimum.

4.2 Look-Ahead Heuristic (LAH)

Subsequently a dynamic program for solving the single-objective problem exactly is introduced. Algorithm 2 determines the shortest path from all nodes to the end node separately for all single criteria. The algorithm is heavily similar to the one of Floyd and Warshall [16], [9].

All objective functions used here are separable and fulfill the terms of Dynamic Programming [1]. The Dynamic Programming algorithm is applied to any criterion of the optimization problem. The output of the Dynamic Program is used to improve the ACO algorithm. The inverted local edge weights $1/f_e^c(e_{ij})$ are no longer used as heuristic information. We set η_{ij}^c to the single-criterion objective function values of the respectively shortest (best) path starting from the node v_i to the end node v_t by using the edge e_{ij} (Look-Ahead Heuristic information, see also [8], [7] for solving parallel path problems).

Formally, a path from a node v_i to a node v_j by using an edge e_{ij} can be defined as in equation (4.2) as $\pi(v_i, e_{ij}, v_t)$.

$$\pi(v_i, e_{ij}, v_t) = \pi(v_i, v_t) \Leftrightarrow e_{ij} \in \pi(v_i, v_t) \tag{4.2}$$

The algorithm introduced here determines all objective function values of the optimal paths between all nodes v_i of the graph and the end node v_t for each criterion. Algorithm 2 shows the optimistic *best-case*-version of Look-Ahead Heuristic information (LAH/bc). Furthermore we have developed two other versions.

The pessimistic *worst-case*-version (LAH/wc) determines all objective function values of all worst solutions from any node v_i to the end node v_t for each criterion. Divergent to lines 8 and 13 in algorithm (2) the calculation of η_{ij}^c and $\pi^*(v_s, v_t)$ results, according to equation (4.3).

$$\begin{aligned} \eta_{ij}^c &\leftarrow f_\pi^c(\pi(v_i, e_{ij}, v_t)) \\ \pi^*(v_s, v_t) &\leftarrow \pi(v_s, e_{sj}, v_t) : \arg \max_{\forall e_{sj} \in E} f_\pi^c(\pi(v_s, e_{sj}, v_t)); \end{aligned} \tag{4.3}$$

Using the *worst-case*-version of the algorithm the determined heuristic values η_{ij}^c describe the least guaranteed objective function value of a (partial) solution from the node v_i to v_t by using the edge e_{ij} regarding to criterion c. The third version of the algorithm determines the average value of the heuristic information calculated by the two other versions and is referred to as *average-case*-version (LAH/ac).

Since the optimal and pessimistic path between start and end node is determined by the heuristic for each criterion, there are known all limitation vectors of the Pareto-front of the problem. Therefore, the diversity level of the initial solutions is improved significantly. Due to the fact that also partial solutions and not only local weights are rated by LAH, this should lead to an improvement of the ACO algorithm.

Algorithm 2 Calculation of Look-Ahead Heuristic values

```
function: MakeHeuristicValues
requires digraph $D$, node $v_s$, node $v_t$
begin
 1: for all criteria $c = 1, \dots, n$ do
 2:     GetPath($D, v_s, v_t, c$)
 3: end for
end function

function: GetPath
requires digraph $D$, node $v_s$, node $v_t$, criterion $c$
begin
 1: if node $v_s$ marked then
 2:     return $\pi^*(v_s, v_t)$
 3: else
 4:     $\pi^*(v_s, v_t) \leftarrow \emptyset$
 5:     for all edges $e_{ij} \in E : i = s$ do
 6:         $\pi(v_i, e_{ij}, v_t) \leftarrow e_{ij} \cup$ GetPath($D$,$v_j$, $v_t$)
 7:         if $v_t \in \pi(v_i, e_{ij}, v_t)$ then
 8:             $\eta_{ij}^c \leftarrow f_\pi^c(\pi(v_i, e_{ij}, v_t))$
 9:         else
10:             $E \leftarrow E \setminus e_{ij}$
11:         end if
12:     end for
13:     $\pi^*(v_s, v_t) \leftarrow$ $\pi(v_s, e_{sj}, v_t)$ : $\arg\min_{\forall e_{sj} \in E} f_\pi^c(\pi(v_s, e_{sj}, v_t))$
14: end if
15: mark $v_s$
16: return $\pi^*(v_s, v_t)$
end function
```

5. EVALUATION

The introduced concepts of the multi-objective ACO algorithm and especially the Look-Ahead Heuristic values are to be evaluated in the following part. Realizing this, instances of the problem were examined which could be solved exactly using a multi-objective Dynamic Program. So the exact solution of the problem is determined for comparison to the approximated fronts by the ACO algorithm. We expect that the behavior of the ACO algorithm applied to much more difficult problem instances is similar.

5.1 Performance Figures

First, the performance figures approximation (apx) and diversity (div) are generated to estimate the applied algorithms. Let Π^A with the elements π^A be the set of approximated solutions determined by the ant algorithm and Π^* with the elements π^* be the set of all Pareto-optimal solutions of the problem.[7] The approximation of a solution $\pi_i^A \in \Pi^A$ is to be described as the shortest vector distance to a solution $\pi_i^* \in \Pi^*$ according to equation (5.2). The diversity describes the shortest vector distance from a solution $\pi_i^* \in \Pi^*$ to a solution $\pi_i^A \in \Pi^A$ according to equation (5.3). As these values are distance values, the aim is to get values near 0.

$$\|f_\pi^c(\pi_i)\|_{[0,1]} = 1 - \frac{f_\pi^c(\pi_i) - f_{\text{ideal}}^c}{f_{\text{neg}}^c - f_\pi^c(\pi_i)} \tag{5.1}$$

[7]If the set of the exact solutions of the problem is unknown, the set of the best known solutions may act as reference set. At this, condition $\Pi^* \succeq \Pi^A$ should be fulfilled. In this article we use only the set of all Pareto-optimal solutions.

$$\mathrm{apx}(\pi_i^A) = \min_{\pi^* \in \Pi^*} \sqrt{\sum_{c=1}^{n} \left(\|f_\pi^c(\pi_i^A)\|_{[0,1]} - \|f_\pi^c(\pi^*)\|_{[0,1]} \right)^2} \tag{5.2}$$

$$\mathrm{div}(\pi_i^*) = \min_{\pi^A \in \Pi^A} \sqrt{\sum_{c=1}^{n} \left(\|f_\pi^c(\pi_i^*)\|_{[0,1]} - \|f_\pi^c(\pi^A)\|_{[0,1]} \right)^2} \tag{5.3}$$

For an exact determination of the vector distance, it is necessary to transform the objective function values of the different criteria in such a way that we get a uniform measurement scale. Therefore, a standardization on the interval between the (positive) ideal point of the problem F_{ideal} and the negative ideal point F_{neg} according to equation (5.1) will be performed.[8] The measures $\overline{\mathrm{apx}}(\Pi^A)$ and $\overline{\mathrm{div}}(\Pi^*)$ depict the normalized average values over all elements of the approximated set of solutions or the reference set of solutions.

$$\overline{\mathrm{apx}}(\Pi^A) = \frac{\sum_{\pi_i^A \in \Pi^A} \mathrm{apx}(\pi_i^A)}{|\Pi^A| \cdot \sqrt{n}} \quad ; \quad \overline{\mathrm{div}}(\Pi^*) = \frac{\sum_{\pi_i^* \in \Pi^*} \mathrm{div}(\pi_i^*)}{|\Pi^*| \cdot \sqrt{n}} \tag{5.4}$$

Due to the compensatory effects, the average values $\overline{\mathrm{div}}(\Pi^*)$ are not quite meaningful. Therefore, we introduce the diversity limit $\mathrm{dlim}(r)$ with $r \in [0,1]$ as an additional measure which describes the number of solutions $\pi_i^* \in \Pi^*$ which at least fulfill $\|\mathrm{div}(\pi_i^*)\|_{[0,1]} \geq r$. Therewith, it describes the number of solutions in Π^* for which there do not exist any solution in Π^A with a vector distance shorter than r. Formally, in equation (5.5) the measure dlim will be defined.

$$\begin{gathered} \mathrm{dlim}(r) = \left|\Pi^{\mathrm{div}}\right| \text{ with} \\ \Pi^{\mathrm{div}} \subseteq \Pi^* : \ \pi_i^* \in \Pi^{\mathrm{div}} \Leftrightarrow \|\mathrm{div}(\pi_i^*)\|_{[0,1]} \geq r \end{gathered} \tag{5.5}$$

The normalized performance figure $\|\mathrm{dlim}(r)\|_{[0,1]}$ describes dlim as a percental ratio from the number of all solutions in the set Π^* according to equation (5.6).

$$\|\mathrm{dlim}(r)\|_{[0,1]} = \frac{\mathrm{dlim}(r)}{|\Pi^*|} \tag{5.6}$$

The values of $\mathrm{dlim}(r)$ can be plotted in a 'curve' for $r = 1, \ldots, R$. This curve describes the structure of the diversity of the approximated solutions in comparison to the reference set Π^*.[9] Furthermore in equation (5.7) a performance figure of the diversity structure dvs is defined which rates the complete structure of diversity [7].

$$\mathrm{dvs}\,(R) = \frac{\sum_{i=1}^{R} i \cdot \|\mathrm{dlim}\,(i/R)\|_{[0,1]}}{R} \tag{5.7}$$

R is a parameter for controlling the precision of the performance figure dvs. In this contribution we use $R = 100$. In dvs the different values for dlim are aggregated by rating high values for r significantly higher than low values for r. With this method bigger distances of single solutions from the reference set to the approximated set in the target area are penalized much more than smaller distances. The lower the value of dvs is, the better the approximated set can be rated regarding the diversity.

5.2 Results

During a parameter test, robust configurations of the multi-objective ant algorithms solely using the Standard Heuristic (inverted edge weights) was determined.

Using the values $q_0 = 0.5$, $\alpha = 1$, $\beta = 3$, $\rho = 0.1$, $\tau_0 = 0.05$, multiple problem instances returned very good results with Ant Colony System. The sampling parameters $\chi_{\mathrm{ants}} = 5$ and $\chi_{\mathrm{colonies}} = 5$ were chosen ($m = w = 15$). The number of $t = 600$ iterations were processed and 9000 solutions were created per optimization run.

It turned out that the parameters $\alpha = 1$, $\beta = 5$, $\rho = 0.01$, $\chi_{\mathrm{ants}} = 3$ and $\chi_{\mathrm{colonies}} = 3$ depict a robust configuration for the Ant System algorithm. Since $m = w = 6$ Ant System has performed $t = 1500$ iterations to create 9000 solutions per optimization run.

Both configurations have also been tested using the Look-Ahead Heuristic values. Table 1[10] shows the results in comparison to each other.

Heuristic	$\overline{\mathrm{apx}}\,(\Pi^A)$	$\overline{\mathrm{div}}\,(\Pi^A)$	dvs (100)	$\lvert\Pi^A\rvert$
		Ant System		
Standard	*0.0440*	*0.2367*	*0.4715*	*67,9*
LAH/wc	0.0255	0.0402	0.1309	104,7
		Ant Colony System		
Standard	0.0231	0.0532	0.3192	106.8
LAH/bc	**0.0161**	0.0247	0.0495	137.7
LAH/ac	0.0169	0.0251	0.0505	135.8
LAH/wc	0.0164	**0.0246**	**0.0467**	**138.0**

Table 1: Average results over 8 problems with an input length ($|E| + |V|$) from 527 to 1408 over 5 runs each.

It is obvious that the results of the Ant System are considerably inferior to the Ant Colony System. Furthermore, all versions of Look-Ahead Heuristic values create significantly better results than the Standard Heuristic. The *worst-case*-version achieves the overall best and most robust results. As the calculation of the Look-Ahead Heuristic values only took some seconds in all problem cases, the developed concept can be approved.

Consecutively, the problem D_1 should be examined more specifically to point out the results. This concerns a problem graph with 415 nodes and 762 edges. The solution space includes a total of approximately $3.95 \cdot 10^{18}$ solutions, the Pareto-front includes a total of 990 non-dominated solutions. Table 2 shows the results for this problem D_1 which are significantly similar to the values from table 1. To secure the convergence of the algorithm, the iterations were raised to $t = 1800$. Thus 18,000 solutions were created per optimization run. This also shows that the usage of the Look-Ahead Heuristic information results in a significant improvement of

[8]The ideal point F_{ideal} usually depicts a non-reachable point in the target area and its coordinates result from the objective function values of the single-criterion optimal solution f_{ideal}^c of all criteria c. The negative ideal point F_{neg} is constructed from the objective function values of single-criterion pessimistic solutions f_{neg}^c. As the problem is solvable exactly for a single criterion the determination of F_{ideal}^c and F_{neg}^c is not problematic.

[9]This is similar to the concept of Lorenz curve [2].

[10]**best value**, *worst value*

the algorithm. Figure 1 shows the Pareto-front of the problem in comparison to the approximated fronts of the Ant algorithms. It is obvious that the approximation of ACS algorithm using LAH is clearly better than the approximation of ACS using the Standard Heuristic information, especially regarding the approximation of the second criterion.

Heuristic	$\overline{\text{apx}}\left(\Pi^A\right)$	$\overline{\text{div}}\left(\Pi^A\right)$	dvs (100)	$\lvert\Pi^A\rvert$
	ANT COLONY SYSTEM			
Standard	*0.0127*	*0.0807*	*0.6998*	*82.6*
LAH/bc	0.0117	0.0176	0.0264	126.6
LAH/ac	0.0123	0.0187	0.0303	**128.8**
LAH/wc	**0.0110**	**0.0168**	**0.0259**	128.6

Table 2: Results of the exemplary problem D_1 (average values from 5 runs each).

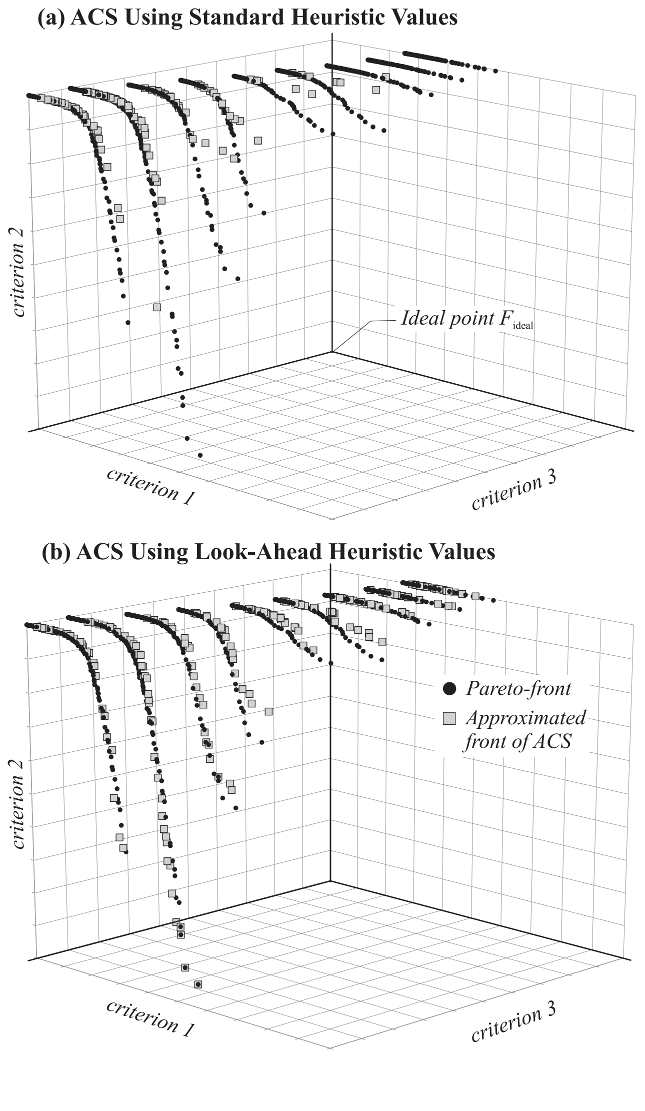

Figure 1: Approximated front of ACS with Standard Heuristic (top) and ACS with LAH/wc (bottom) in comparison to the exact Pareto-front.

Figure 2 shows the curves of the diversity structure with the percental diversity limits $\|\text{dlim}(r)\|_{[0,1]}$ for all $r = i/R$ with $i = 1, \ldots, 100$ and $R = 100$ by using ACS with the Standard Heuristic and LAH. The values of the curve of ACS/LAH decrease significantly faster and earlier than the values of the other curve. Therewith a solution in the approximated set exists for a lot more solutions of the reference set.

Figure 3 shows the development of the approximation, diversity and the size of the approximated front in relation to the progress of the iterations. This once more shows that the use of the Look-Ahead Heuristic information achieves a significant improvement, especially regarding the development of diversity and front size.

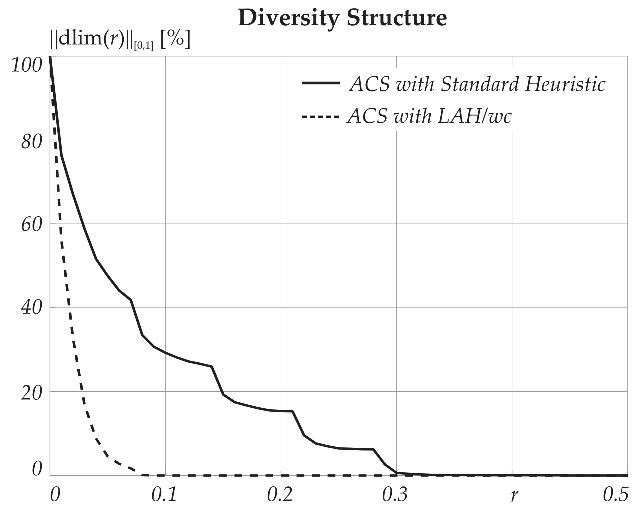

Figure 2: Curve of diversity structure of the algorithm versions applied to D_1 with dlim(r) for all $r = i/R$ with $i = 1, \ldots, 100$ and $R = 100$ (average values from 5 runs each).

6. CONCLUSIONS

This article demonstrates that the hybrid approach of the ACO Meta-heuristic and Dynamic Programming with the help of the Look-Ahead Heuristic information values can lead to significant improvements of the results.

The usability is based on the fact that the single-objective version of the optimization problem is solvable in an appropriate time.

The discussed concept can not only be applied to shortest path problems but also to any other problem with a similar structure. The only important requirement is that the solution is possibly using Ant Colony Optimization and the objective functions of the criteria fulfill the terms of Dynamic Programming.

7. REFERENCES

[1] R. Bellman. *Dynamic programming.* Princeton Univ. Press, Princeton, 1957.

[2] G. Bol. *Deskriptive Statistik.* Oldenbourg Verlag, München, 1998.

[3] C. Coello Coello, V. Veldhuizen, D. Allen, and G. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*, volume 5 of *Genetic algorithms and evolutionary computation*. Kluwer, Acad. Publ, New York, NY, 2002.

[4] E. Dijkstra. A note on two problems in connexion with graphs. Number 1, pages 269–271. Mathematisch Centrum, Amsterdam, The Netherlands, 1959.

[5] M. Dorigo and L. Gambardella. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, 1997.

[6] M. Dorigo, V. Maniezzo, and A. Colorni. The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 26(1):1–13, 1996.

[7] M. Fischer. *Partnerauswahl in Netzwerken – Ein mehrkriterieller Optimierungsansatz zur Bestimmung effizienter Netzkonfigurationen basierend auf Ant Colony Optimization*. Verlag Dr. Kovač, Hamburg, 2008.

[8] M. Fischer, S. Häckel, and J. Kaminski. Improving Ant Colony Optimization for solving Multiobjective Shortest Path Problems by Using the new Look-Ahead-Heuristic-Concept. In *Proceedings of the 22th International Conference on CAD/CAM, Robotics and Factories of the Future (CARS & FOF 2006)*, Vellore (India), July 19.-22. 2006.

[9] R. Floyd. Algorithm 97: Shortest Path. *Communications of the ACM*, 5(6):345, 1962.

[10] M. Guntsch. *Ant algorithms in stochastic and multi-criteria environments*. PhD thesis, Universität Karlsruhe (TH), 2004.

[11] P. Hansen. Bicriterion Path Problems. In G. Fandel and T. Gal, editors, *Multiple Criteria Decision Making Theory and Application*, Lecture notes in economics and mathematical systems 177, pages 109–127, Berlin, Heidelberg, 1980. Springer.

[12] S. Iredi, D. Merkle, and M. Middendorf. Bi-Criterion Optimization with Multi Colony Ant Algorithms. In E. Zitzler et al., editor, *Proceedings of the First International Conference on Evolutionary Multi-Criterion-Optimization (EMO'01), Lecture Notes on Computer Science 1993*, pages 359–372, Zürich, 2001. Springer.

[13] J. Jahn. *Vector Optimization - Theory, Applications, and Extensions*. Springer, London, 2004.

[14] K. Miettinen. *Nonlinear Multiobjective Optimization*, volume 12 of *International series in operations research und management science*. Kluwer, Acad. Publ, Boston, MA, 1999.

[15] P. Serafini. Some considerations about computational complexity for multi objective combinatorial problems. In J. Jahn and W. Krabs, editors, *Recent Advances and Historical Develpment of Vector Optimization*, Lecture notes in economics and mathematical systems 294, page 405, Berlin, 1987. Springer.

[16] S. Warshall. A Theorem on Boolean Matrices. *Journal of the ACM*, 9(1):11–12, 1962.

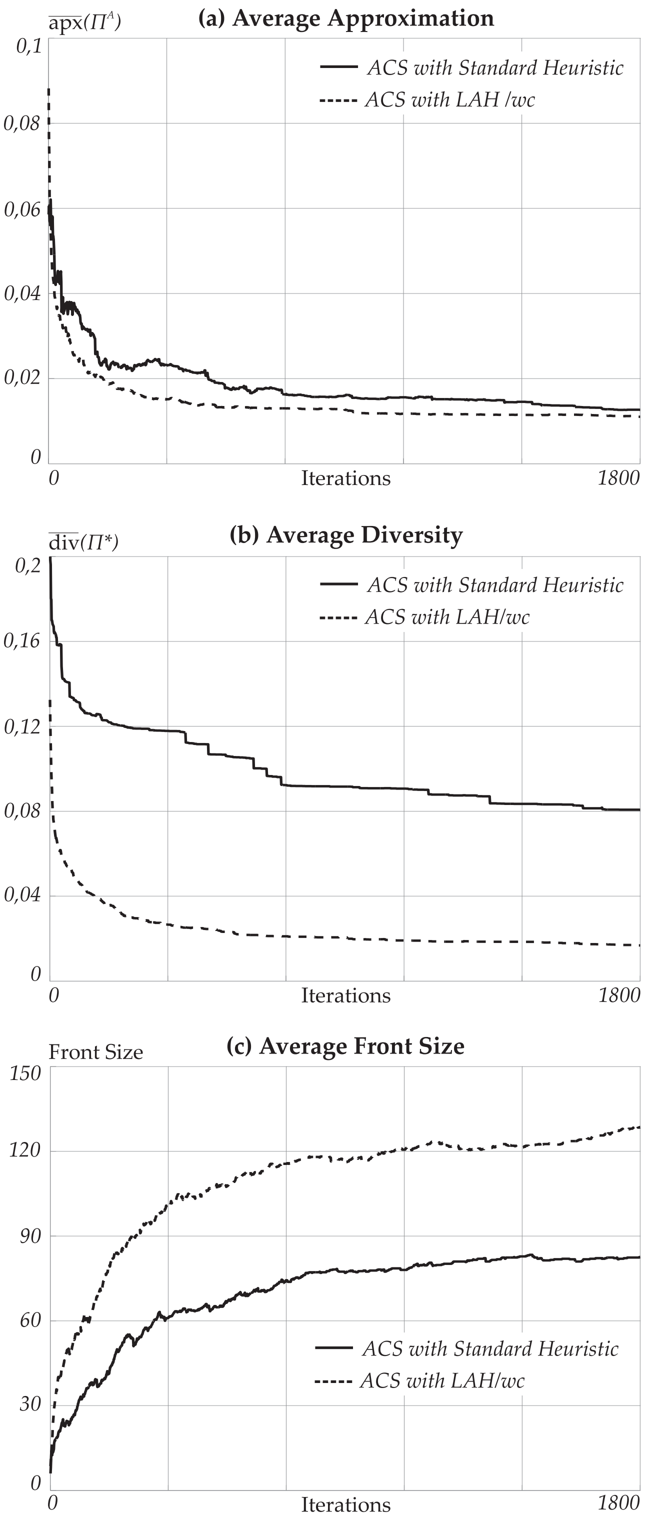

Figure 3: Development of the approximation, diversity and front size for D_1 within the progress of the iterations (average values from 5 runs each).